(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

HYPERMEDIA AND DIGITAL OPTICAL MEDIA TECHNOLOGIES AS APPLIED TO A PROTOTYPE GEOGRAPHIC AND THREAT RECOGNITION (GEOTREC) TRAINING AND REFERENCE TOOL

by

Wayne F. Sweitzer

March 1990

Thesis Advisor: **Barry A. Frew**

Approved for public release; distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4 PERFORMING ORGANIZATION REPORT NUMBER(S)		5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL *(If applicable)* Code 37	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000		7b ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (*City, State, and ZIP Code*)		10 SOURCE OF FUNDING NUMBERS	
		Program Element No	Project No / Task No / Work Unit Accession Number

11 TITLE (*Include Security Classification*)

HYPERMEDIA AND DIGITAL OPTICAL MEDIA TECHNOLOGIES AS APPLIED TO A PROTOTYPE GEOGRAPHIC AND THREAT RECOGNITION (GEOTREC) TRAINING AND REFERENCE TOOL (UNCLASSIFIED)

12 PERSONAL AUTHOR(S) Sweitzer, Wayne F.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*year, month, day*)	15 PAGE COUNT
Master's Thesis	From To	1990 March	174

16 SUPPLEMENTARY NOTATION

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	HYPERMEDIA, HYPERTEXT, DIGITAL OPTICAL MEDIA, THREAT RECOGNITION

19 ABSTRACT (*continue on reverse if necessary and identify by block number*)

Threat recognition and geographical training are fundamental parts of the requisite knowledge base for a large number of naval personnel who are assigned to operational or operations-oriented support billets. Yet readiness in these areas is often lacking, in large part due to the paucity of readily available, motivational instruction tools.

This thesis explores major issues involved in integrating two emerging technologies, hypermedia and digital optical media (DOM), in the context of developing a prototype of just such an application: the GEOgraphic and Threat RECognition (GEOTREC) training and reference tool. The hypermedia software package used to develop the GEOTREC prototype, Hyperdoc version 1.12, gives evidence of the maturation yet needed in the integration of hypermedia and DOM technologies in application authoring tools.

This thesis recommends the development of a system at least somewhat analogous to the GEOTREC prototype. Such a tool, using both hypermedia and DOM, would not only provide an enjoyable, intuitive, yet challenging way to foster multi-sensory learning, but also a quick, powerful, and easy-to-use reference to geographical and threat information needed for a myriad of operational scenarios.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
[X] UNCLASSIFIED/UNLIMITED [X] SAME AS REPORT [ ] DTIC USERS	UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Barry A. Frew	22b TELEPHONE (*Include Area code*) (408) 646-2391	22c OFFICE SYMBOL Code 54FW

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Approved for public release; distribution is unlimited.

Hypermedia And Digital Optical Media Technologies
As Applied To A Prototype
Geographic And Threat Recognition (GEOTREC)
Training And Reference Tool

by

Wayne F. Sweitzer
Lieutenant, United States Navy
B.A., Gordon College, 1980

Submitted in partial fulfillment
of the requirements for the degree of
MASTER OF SCIENCE IN INFORMATION SYSTEMS
from the
NAVAL POSTGRADUATE SCHOOL
March 1990

Author: _____
Wayne F. Sweitzer

Approved by: _____
Barry A. Frew, Thesis Advisor

_____
Dr. Vincent Y. Lum, Second Reader

_____
David Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

Threat recognition and geographical training are fundamental parts of the requisite knowledge base for a large number of naval personnel who are assigned to operational or operations-oriented support billets. Yet readiness in these areas is often lacking, in large part due to the paucity of readily available, motivational instruction tools.

This thesis explores major issues involved in integrating two emerging technologies, *hypermedia* and *digital optical media (DOM)*, in the context of developing a prototype of just such an application: the GEOgraphic and Threat RECognition (GEOTREC) training and reference tool. The hypermedia software package used to develop the GEOTREC prototype, Hyperdoc version 1.12, gives evidence of the maturation yet needed in the integration of hypermedia and DOM technologies in application authoring tools.

This thesis recommends the development of a system at least somewhat analogous to the GEOTREC prototype. Such a tool, using both hypermedia and DOM, would not only provide an enjoyable, intuitive, yet challenging way to foster multi-sensory learning, but also a quick, powerful, and easy-to-use reference to geographical and threat information needed for a myriad of operational scenarios.

TABLE OF CONTENTS

I. INTRODUCTION

Training for U.S. Navy flight crews, Combat Information Center teams, ship's lookouts, and Intelligence Center watchstanders in the areas of threat recognition and geographic familiarity is continually in need of enhancement. Threat recognition and geographical training are fundamental parts of the requisite knowledge base for these and other naval personnel who are assigned to operational or operationally-oriented support billets.

And yet readiness in these areas is often lacking, in large part due to the paucity of readily available, motivational instruction tools. In addition, there is no single, effective, quick-reference source to which these personnel can go for timely geographic and threat recognition information in an operational environment.

This thesis will look at a possible computer-based solution to these problem areas. It will study the use and integration of two emerging technologies, *hypermedia* and *digital optical media (DOM)*, in the context of a specific operational application. That application, the GEOgraphic and Threat RECognition (GEOTREC) training and reference tool, attempts to provide a powerful, easy-to-use, didactic system to address these training and readiness shortcomings. This thesis will focus on the development of a prototype GEOTREC, using a specific hypermedia authoring software package, GECI International's *Hyperdoc*, version 1.12.

This research represents an excursion into areas of technology and application about which much has already been written. It does not necessarily attempt to sail through uncharted waters, embody technological breakthroughs, or make dazzling discoveries. Rather, it explores the utilization of well-documented emerging technologies to meet operational requirements that exist now in today's Navy.

In order to do this, the thesis starts by examining the nature of these emerging and developing technologies and the major issues concerning their utilization and integration. First, Chapter II surveys hypertext and its logical extension, hypermedia. The chapter attempts to explain in lay terms the concepts and principles behind hypermedia, and to articulate its benefits over the linear presentation of information for certain applications. Following this, Chapter III gives a brief overview of digital optical media (DOM), describing the capabilities and utility of this technology.

Next, Chapter IV describes an application, namely the GEOgraphic and Threat RECognition (GEOTREC) training and reference tool, which integrates hypermedia and DOM into an ideal solution for the training and readiness shortcomings. This ideal GEOTREC was used as the model after which the prototype application was fashioned. Appendices A and B provide evidence of this GEOTREC prototype by respectively capturing a simple system demonstration, and providing the code generated in the system's development. The prototype GEOTREC served as the basis of this thesis, providing a focused context for the study of hypermedia and DOM technologies.

The difficulties encountered in the development of the GEOTREC prototype are outlined in Chapter V. The chapter delineates the major obstacles, limitations and difficulties in developing and implementing these technologies in an application of this nature using the hypermedia authoring tools provided by Hyperdoc 1.12. Finally, Chapter VI offers the most salient conclusions and recommendations which have been engendered by the application of these technologies to the development of the GEOTREC prototype.

II. HYPERTEXT/HYPERMEDIA

A. WHAT IS HYPERTEXT ?

Hypertext is a term encompassing a myriad of concepts, ranging from cognitive psychology theory to computer technology. In its broadest context, hypertext is based on a paradigm of the human thought process called *associationalism* which asserts that the mind functions in a series of associative leaps from one idea to another [Ref. 1].

> From the earliest literature on hypertext (e.g., the July 1945 *Atlantic Monthly* article "As We May Think" by Vannevar Bush), much emphasis has been placed on the idea that hypertext structures data in a manner similar to human cognition: in particular, the organization of memory as an [sic.] semantic network in which concepts are linked together by associations. ([Ref. 1], p. 129)

In the more narrow context of computer technology, hypertext has been defined in The Computer Glossary as: "A technique that links information together. Words are invisibly linked to other words or explanations. For example, by pointing to a key word in a sentence and selecting it, the linkage is activated and the associated information is revealed." ([Ref. 2], Doc.#1522)

A synthesis of these diverse hypertext contexts is provided in the introduction to the documentation for "HyperShell", a shareware hypertext software package. In that documentation, hypertext is aptly described as the application of computer technology to the presentation of information occurring in a *non-linear* form. The difference between *linear* information and non-linear information is characterized by the difference between a novel and a reference book. With a novel, a reader starts at page one and reads straight through to the

end, in a linear fashion. However, when a reader approaches an encyclopedia, dictionary, or other such tutorial book, he looks things up in an index, or table of contents, and follows references in certain parts of that book to other chapters, or even other books. It is this latter, non-linear type of literature and information for which computer-based hypertext applications are best suited. ([Ref. 3], p. 2)

It is not at all easy to embody a full description of the hypertext concept in simple definitions. The following quotes are intended to provide an expanded understanding of hypertext:

> In a hypertext document, marked words or commands are doorways to information in other parts of the document, in other documents, or even in other applications. Reading a hypertext document is like playing hopscotch with ideas. Since each area or block of information may be linked to several others, you can jump around wherever your fancy takes you. That's both the strength and weakness of hypertext. ([Ref. 4], p.12)

> Think of a collapsible and expandable file alive with words, sound, animation, and photographs; that will someday become your basic hypertext document. Today,...existing hypertext programs let you affix electronic threads between words or pictures so you can grab a fistful of ideas in one place, in one sitting. ([Ref. 5], p. 76)

> The concepts of Hypertext are very old; they have been transferred to computer systems since the [19]60's.... Originally intended to manage arbitrarily linked text segments, Hypertext has been extended to manage images and sound as well ("Hypermedia"). ([Ref. 6], p. 18)

B. WHAT IS HYPERMEDIA ?

The latter two quotations mention more than just text; they allude to the assimilation of other media into the hypertext arena, a technique often called *hypermedia* to indicate the expanded, multimedia scope. Once again, from The Computer Glossary: "Hypermedia refers to the use of data, text, graphics, video and voice as elements in a Hypertext system. All the various forms of information are linked together so that a user can easily move from one to

another." ([Ref. 2], Doc.#1522) Music, audio and other signals can be added to the above list.

Although the term "hypertext" in its strictest definition pertains only to text, it is frequently used today to refer to the broader spectrum of information-linking in a variety of media, thereby encompassing "hypermedia" as well. Except where indicated, this paper will use "hypertext" in this larger, multimedia sense, treating the terms "hypertext" and "hypermedia" as virtually interchangeable.

C. WHAT CAN HYPERMEDIA DO ?

A graphically descriptive scenario for the use of hypertext is given in an article in the July 1989 issue of *PC-Computing* magazine. It asks the reader to imagine for a moment reading that magazine on a personal computer (PC) instead of on paper. When the "magazine file" is opened, the table of contents appears. Browsing through the headlines, the reader notices an article about hypertext, uses the mouse to place the cursor on the article's title, and clicks on it. The article appears on the screen, describing how a major oil company is using hypermedia to keep track of its drilling operations. To find out more about how the company is using this new method of accessing information, the reader clicks on the oil company's name. This activates an attached videodisc player, starting a video presentation, complete with music, highlighting the benefits of hypermedia to the company's information needs. Following the video segment, the reader wants to learn where the company obtained their hypermedia system. A click on the name of the hypermedia vendor, and the screen displays an address, phone number, and product specifications. Just then, the doorbell rings, so the reader places a marker, and closes the magazine file. Thus, the next time he opens the electronic

magazine, the page where he left off is immediately available. ([Ref. 5], pp. 75-76)

Multimedia applications of hypertext can provide an enormous amount of information to a user in a wide spectrum of vivid, captivating forms:

> In the context of a [hypermedia] video presentation, say, of an anatomy lesson, the collar bone can be displayed concurrently with data; click on a part of the collar bone on-screen and a short, technical description could appear. Other programs can use running text along with a videodisc presentation: On a two-monitor system, you see the film running by on the video display, while commentary scrolls along with it on your [computer's] screen. You stop the videodisc, click on an annotation button, and more information comes up on what you're viewing. In the digital world, sights become sounds, sounds become words, the word duck becomes a mallard flying across your screen. ([Ref. 7], p.100)

Clearly, hypermedia applications provide capabilities for information retrieval never before possible with paper documents. Users can access multimedia information in the sequence, volume and format that best suits their needs at the time they access the information. In addition, they can change their access strategy each time they need to refer to that information. ([Ref. 1], p. 22)

In addition to providing *users* with new capabilities, the *authors*, or creators, of hypertext applications can easily write (author) these applications for a myriad of multimedia uses. These include, but are by no means limited to ([Ref. 8], p.28):

- Education and training;
- Entertainment;
- Travel;
- Multi-language applications;
- Real estate;

- Retail kiosks and information booths;
- Landscaping, design and decorating.

A March, 1989, article in *Digital Review* magazine reports how corporate and commercial designers are using HyperCard, a hypertext product from Apple Computer, to create impressive multimedia projects. These applications include live-action video and still images stored on laser disks, high-quality audio tracks for direct output from a CD player, and high-resolution color images, music and speech stored as computer data. The developers are also using color scanners, music synthesizers, touch screens and other state-of-the-art input and output devices. Non-multimedia HyperCard applications are also popular with hypertext authors, and range from indexing large amounts of CD-ROM data to developing interactive tutorials and reference works. ([Ref. 9], p.19)

The capabilities of hypermedia software have been largely unexplored until recent years. The reason for this reluctance in the past has largely been commercial viability: computer technology had not progressed to the point where such capabilities were affordable to a wide range of users in the marketplace. Even though advantages of hypertext have been evident for several decades, widespread interest in hypertext has been delayed until the supporting technology was cheap and readily available. ([Ref. 10], p.32)

Today, however, the maturation of several of these supporting technologies appear to be converging to take advantage of hypertext's offerings, particularly in the microcomputer arena:

- Computer graphics and the Graphical User Interface (GUI);

- Digital Optical Media technology (e.g., CD-ROM, videodisc, WORM);
- PC imaging systems which digitize photos, documents, etc.;
- Optical Character Recognition (OCR).

However, this "technology lag" may not tell the whole story of hypertext's slow start. Many experts feel that something more than just technology had to change, namely computer users themselves. Today's computer users more easily accept the role of the computer as a tool for processing ideas, words, and symbols, in addition to numbers and mere data. Their vision for the use of the computer as a vehicle of multimedia interhuman communication has greatly expanded beyond that of computer users of even just half a decade ago. ([Ref. 10], p.32)

D. WHAT CAN HYPERMEDIA NOT DO ?

While hypermedia provides powerful tools for information retrieval and manipulation, it does *not* represent a breakthrough which will supplant traditional database models and systems. "HyperCard can't do everything. It's a great prototyping tool for indexing, creating interactive applications and training programs, and for multimedia work. But don't try to use it as a substitute for a full-featured database program...." ([Ref. 9], p.19) The warning is applicable to all hypermedia software.

Perhaps the most significant limitation of hypermedia applications is that they rely on databases which are pre-structured by the application author for one specific application. Such a database does not lend itself to easy use by other applications without a considerable amount of adaptation and reorganization (such

as creating a new "hyperdocument"; see section II.E. for more on hyperdocuments). This is especially true when multiple applications need to access the same database simultaneously.

Moreover, hypermedia applications cannot satisfy a wide range of ad hoc queries. Take, for example, a hypertext cookbook application, using a database consisting of, among other things, textual recipes and digitized photographs of the prepared foods. A user could hypothetically navigate through the "hyper-cookbook" as follows:

1. opening screen provides a breakdown by categories (e.g., "Appetizers", "Desserts", "Meats & Poultry", "Breads", etc.) -- user selects "Desserts";

2. second screen provides dessert types (e.g., "Cookies", "Cakes", "Puddings", "Pies", etc.) -- user selects "Pies";

3. third screen provides specific pie selections (e.g., "Apple Pie", "Pumpkin Pie", "Boston Creme Pie", "Pecan Pie", etc.) -- user selects "Apple Pie";

4. fourth screen shows the recipe, with certain keywords highlighted (e.g., "apple", "crust", etc.), and a menu of choices which are standard on every recipe screen, such as "Photo", "Measurement Conversions", "Microwave Version", and the like;

5. the user then selects the highlighted word "apple" to get a more detailed explanation of apples (e.g., apple types, such as "Cortland", "MacIntosh", "Granny Smith", etc.), with options like: displaying an image of each apple type on the screen; providing information as to which are best for baking, making cider, or eating raw; or even a short musical video clip of the adventures of Johnny Appleseed (for the kids, of course);

6. the user then returns to the "Apple Pie" recipe and follows the instructions, selecting "Measurement Conversions" and other such help items from the standard menu as needed;

7. as the pie is baking, the user selects "Photo" from the standard menu, and then dreams about how the one in the oven is going to look even better than the image now on-screen...

While this hypothetical application obviously gives the user access to a great deal of information in an associative, intuitive manner, it does NOT permit complex, ad hoc, relational queries, such as:

(a) give me the names of all recipes which:
USE -- main ingredients: "apples",
AND -- spices: "cinnamon AND nutmeg",
BUT NOT -- main ingredients: "sugar",
AND HAVE -- calories per serving: "< 300";

(b) list all "Photos" which:
SHOW -- foreground elements: "apple pie OR pumpkin pie OR boston creme pie",
AND -- background elements: "silver pie server AND cup of steaming coffee".

In order to provide such functionality, a broad database built on the relational model is required. A major obstacle involved in developing and managing such a database is the unstructured nature of data types such as image, sound and signal. Research is currently underway at the Naval Postgraduate School in Monterey, California, into the mammoth task of incorporating the multimedia-handling features of hypermedia into the relational database model. One of the goals of this effort is to extend relational database management systems (DBMS) to manage multimedia databases, and to support numerous multimedia applications simultaneously. [Ref.s 6, 11, and 12]

While not intended to supersede traditional database systems, most hypermedia software packages can act as a "front-end" to a relational database by allowing the user to link to a DBMS when appropriate. For example, if the user's query had to do with an automobile clutch, then the he would click on the graphic representation of the clutch on a diagram. A menu would pop up asking what he

wanted to find out about the clutch. If it was how to change a clutch he would be routed to various static levels of explanation. However, if he wanted to find out dynamic information, such as how many clutches were in stock, ordered primarily by price and secondarily by part number, the hypermedia link to the DBMS would access a relational database. ([Ref. 13], p.1)

In addition to the above limitations, hypermedia does NOT provide a universal, all-encompassing solution to information storage, retrieval and presentation. For instance, "...not all texts are suitable for hypertext representation....[In fact,] experience suggests that if the document is closely interwoven through rhetorical devices, then decomposition into chunks and links[1] will be difficult, with lose [sic.] of information and confusion of meaning a potential result. For some documents, this conversion is either impossible or not desirable because it destroys the subtle interconnections of theme, argument, metaphor, and word choice." ([Ref. 1], p.63).

E. HOW IS HYPERMEDIA IMPLEMENTED ?

The above discussion of hypermedia's definition, capabilities and limitations would be incomplete without at least a brief exploration of how hypermedia is actually implemented in applications.

As previously indicated, the creator of a hypermedia application is often referred to as its author. The author designs an application using the *node* as a basic building block. Each node consists of a single idea or concept. As the author thinks of new ideas, he can develop them into nodes and link them to

---

[1] Section II.E. will further describe these hypertext chunks, or nodes, and the links that connect them.

existing nodes, or he can leave them temporarily isolated if making such associations would be premature. ([Ref. 10], p.38)

The expertise required for authoring, from casual user to experienced programmer, depends both upon the nature of the application and upon the hypermedia package being used. There are many hypermedia software packages presently available for all levels of computing, ranging from text-handling-only to full multimedia. Host systems range from Macintosh or IBM-compatible microcomputers to large mainframes. A broad sampling of existing hypertext packages includes:

- HyperCard (Apple Computer -- for Mac/OS)
- SuperCard (Silicon Beach -- for Mac/OS)
- Guide (Owl International -- for Mac/OS, MSDOS)
- HyperShell (shareware by Text Technology -- for MSDOS)
- KnowledgePro (Knowledge Garden -- for MSDOS)
- LinkWay (IBM -- for MSDOS, OS/2)
- Hyperdoc (GECI International -- for MSDOS, UNIX)
- Topic (Verity -- for MSDOS, OS/2, VAX/VMS, and UNIX)
- Intermedia (Brown University -- for UNIX, Mac/OS)

There are, at a minimum, three basic components of any hypertext application:

1. a means for the user to interact with the application, normally a graphical user interface (GUI), i.e., menu-driven and mouse-oriented, as opposed to command-line-entry;

2. a database consisting of elements, or nodes, of relevant information; and,

3. active cross-references, or links, between the nodes of that database.

The combination of nodes and their connecting links form a hypertext network, commonly referred to as a *hyperdocument.* This hyperdocument is the file or collection of files which serves as the blueprint of the hypertext application, giving the online information its structure and connectivity. ([Ref. 1], p. 61)

Each hypertext package provides a variety of tools with which the author can build an application. These range from embedded text editors or word processors (or the ability to import files created with separate word processing software), to a myriad of menu-driven tools, to the incorporation of macro-commands (macros) and even source code which is written in third and/or fourth generation programming languages that the package will compile. New authoring tools are continually being developed:

> A paradoxical symbiosis exists in all of this. Apple and other [hypertext software package] developers cannot provide users with tools until users themselves have determined what it is they want to do, while users need tools to discover what's possible. As with all innovation, it's a process of education. Providing more linking capabilities will lead us to providing users with additional tools to better understand more concepts. How do people work with hypertext, hyperauthoring? The number of people that have reasonable answers to this stuff is very limited... ([Ref. 14], p.18)

Using appropriate tools, an author begins by establishing, in the first node of the hyperdocument, the general subject areas in the application's database. This acts much the same as a table of contents in a book, and will normally constitute information and selection options that will appear to the user on the opening screen, or frame. From these starting points, the author can build a network of links between associated nodes as he wishes, both downward, as in a hierarchical tree structure, and/or sideways, as in a network data model. In this way, the

application's user can select and follow whichever series of links he chooses within constraints of options specified by the author in the hyperdocument.

These links are not limited to tying nodes of the same hyperdocument to one another. One of the more powerful features of many hypertext packages is the so-called *hot link*. A standard hypertext link connects information within the same application; by contrast, a hot link allows the user to access information in other applications. Hot links could be used to connect such diverse applications as, for example, a Lotus 1-2-3 spreadsheet, a video presentation, an animation sequence, and, as mentioned before, a relational database.

Full capabilities of hot links are still under development. For instance, a hot link can be forged to a specific file, but cannot be defined to link to a specific place in that file. The user cannot yet use hot links to jump to a specific cell in a spreadsheet, or have a variable in a hypertext report automatically updated when that same variable is changed in the table of a relational database package such as Oracle, Ingres, or Paradox. This capability is coming, however; experts in the field speculate that, within the next five years, almost all hypertext programs will allow the author to easily create such sophisticated hot links. ([Ref. 4], p.12)

In addition to these components, hypertext applications often need a means of keeping track of where the user is "located" in this network of linked ideas at any given time. If the application is anything but very small, the user can easily get disoriented, particularly if he wants to return to an idea (a screen, menu, node, frame, etc.) previously encountered. Most applications allow the user to navigate around the hyperdocument using a "road map", known as a *browser*, which displays the network graphically, indicating the user's position.

The browser is an important component of the larger, more complex hypertext systems. Its display of some or all of the hyperdocument as a graph provides "an important measure of contextual and spatial cues to supplement the user's model of which nodes he is viewing and how they are related to each other and their neighbors in the graph." ([Ref. 10, p.19)

Due to the nature of hypermedia, authors can develop applications with broad utility and appeal. Because so many forms and sources of information can be linked together, an application can be easily written for a wide range of expertise without confusing the novice or boring the expert. Each reader can pursue a topic to the depth that is most appropriate to his or her needs. This allows the hypertext author to use high-level concepts terminology; novices can click on any unfamiliar terms in order to learn about them, while more knowledgeable readers can move ahead to more advanced topics. ([Ref. 4], p.41, and [Ref. 1], p. 124)

Take, for example, an application dealing with aviation-related training. The key personalities, aerodynamic concepts, and acronyms of the trade could simply be mentioned without further description for the sake of more advanced users. At the same time, other readers could click on those key terms and follow the links to the data in related nodes if they need further explanations, help or background material.

In summary, hypermedia has moved in just the past five years from largely theoretical and experimental concepts to highly useful and commercially available real-world applications. Within the bounds of inherent limitations, some of which

have been discussed in this chapter, its potential appears to be enormous and could strongly impact the way users interact with information:

> ...the greater sense of control over the reading process may produce increased [reader] involvement and the desire to read more. In the same way that computer games can be very absorbing because of the high level of interactivity, hypertext databases may be very engaging too. ([Ref. 1], p. 129)

III. DIGITAL OPTICAL MEDIA

A. BACKGROUND

Binary digital data -- information encoded into zeros and ones for computer processing -- can be stored in many forms and on a variety of media. In the late 1960s and 1970s, magnetic tape, somewhat similar to that used on standard tape recorders, was perhaps the most prominent medium for mass storage of digital data. In the late 1970s and 1980s, the use of magnetic disk became widespread. A different category of media, one that is non-magnetic, has emerged from research laboratories and into increasingly greater public use: Digital Optical Media (DOM).

Digital Optical Media (DOM) are also know as *optical laser* (or even laser optical) *media*. (For simplicity and ease of reference, the remainder of this paper will use the "DOM" abbreviation to refer to this category of media, and will use it as if singular, even though the last term in DOM, "Media", is actually plural). Whichever terms are used, they refer to any media in which the information is first converted to binary digital form, and then into its recorded form by use of an optical *laser* (Light Amplification by Stimulated Emission of Radiation). To read the data, the process is reversed: the recording is read by a low-powered laser, converted into binary digital form, and then into whatever form is appropriate for an application. ([Ref. 15], p.71)

The first commercially prosperous DOM was the audio Compact Disc, or CD, the familiar shiny-silver disks, just under five inches in diameter, which are

quickly replacing vinyl record albums in the music recording industry. Somewhat less successful has been the Compact Disc Video (CDV), more frequently called a videodisc or laser disk. These look similar to CDs, but are 12 inches in diameter, and normally store a feature-length movie as well as its soundtrack in digital form. Both of these DOM have been commercially available in meaningful quantities since the early to mid-1980s.

Only relatively recently has the optical disk technology behind CD and videodisc been applied with a significant degree of success to the computer field, outside of the entertainment industry. DOM was initially applied there to meet largely static mass data storage requirements such as those found in libraries, archives, technical manuals, legal records, and the like. Today, however, the "buzzword" application for DOM is *multimedia*. As previously described in Chapter Two, this multimedia (sometimes called Digital Optical MultiMedia, or DOMM) consists of data, text, sound, images, graphics and video all stored in digital format, and on the same disk.

B. GENERAL ADVANTAGES OF DIGITAL OPTICAL MEDIA (DOM)

Part of the attraction of using DOM for both mass storage and multimedia applications is that it "delivers many bytes in a small volume (high storage density), attains low error rates (high fidelity), and can be replicated rapidly in large numbers at a modest cost (economy)." ([Ref. 15], p.68)

Consider storage capacity. Each CD-ROM (Compact Disc-Read Only Memory) -- currently, the most common form of DOM -- has an effective track density of 16,000 tracks per inch, with a mere 1.6 micrometers separating each track. In contrast, 5.25 inch floppy disks, despite being larger than CD-ROMs,

have a density of only 96 tracks per inch. ([Ref. 16], p.27) Due to this density, a CD-ROM can store up to 680 megabytes of digital data -- over 560 times as much data as a high density 5.25 inch floppy disk. ([Ref. 17], p.17) Such gargantuan storage capacities are shared by other types of DOM, as well: Eastman Kodak Company manufactures a 14 inch optical disk that can hold 6.8 gigabytes (i.e., 6.8 billion bytes) of data; if the data on just one such disk were text, and that text was printed on paper, nearly 275 file cabinets would be required for storage space! ([Ref. 16], p.25)

Another clear advantage of DOM is that it is much more durable than magnetic media. Read-only optical disks consist of a base, or shell, made of a light, durable plastic which is covered with a thin reflective layer (normally aluminum)[2]. The data are represented on the reflective layer by a series of pits (depressions) and lands (the flat surfaces between pits). When the disk is rotated, and then illuminated, or read, by a low-powered laser, the lands reflect back more light than the pits. A change in the reflected light caused by movement from one level to the other (pit-to-land, or land-to-pit) is translated as a binary one; no change in the reflected light (pit-to-pit, or land-to-land) represents a binary zero. For illustration, a series of binary digits such as "1110 0100" would be written to the disk as: "land-to-pit, pit-to-land, land-to-pit, pit-to-pit, pit-to-pit, pit-to-land, land-to-land, land-to-land". Once the data to be stored are transcribed onto the disk as pits and lands, the disk is coated with a hard, resilient, clear lacquer.

---

[2] This does not fully apply to WORM or M-O disks, as will be discussed later in this chapter.

Due to this durable lacquer coating, DOM is generally more secure than magnetic media. Because altering the data on a read-only optical disk or a WORM disk requires some form of physical destruction of the medium, the stored data cannot be easily tampered with, intentionally or unwittingly damaged, or inadvertently erased (an M-O disk provides the same data security, when not in the presence of the high-intensity write laser contained in the M-O drive). And because there is no physical contact between the laser and the disk, the medium does not get worn with time and usage.

In addition, DOM is virtually impervious to magnetic fields, normal climatic conditions or changes (e.g., high humidity, reasonably cold temperatures, etc.), and even some dust, dirt, or minor scratches. By contrast, magnetic media are more or less vulnerable to all of these. Finally, most estimates indicate that optical disks should last at least 30 years with normal usage, and may even last up to a full century. Most magnetic media, on the other hand, are considered reliable for just three to five years.

C. DISADVANTAGES OF DIGITAL OPTICAL MEDIA (DOM)

DOM is clearly not appropriate for every application. Because of the permanent nature of read-only optical disks, they should not be used for data that are frequently changing and in need of updates; WORM and M-O disks can be used in such applications, but in many cases a hard disk or Bernoulli box would work better.

In addition, errors introduced prior to the manufacture of a read-only optical disk are as permanent as the media. Such errors could include anything from typographical errors in the text of a technical publication, to incorrect colors in

graphical images of the national flags of past World Cup Soccer champions. Although the chances of introducing errors during the actual optical disk production are minuscule[3], complete and thorough scrutiny of the data during pre-mastering is essential to ensure an error-free final product.

Another drawback for DOM is its inherently slow operating speed relative to magnetic media such as hard disks. A CD-ROM, for instance, has an average data transfer rate of 150 kilobytes per second, as opposed to a typical hard disk with data transfer rates of 700 kilobytes per second or more. In addition, CD-ROM access speeds can average from one to ten seconds, as opposed to hard disks whose access times are measured in milliseconds. Because of these slower speeds, DOM is generally not useful for truly "real-time" applications. ([Ref. 17], p.17; and, [Ref. 18], p.14)

D. MAJOR TYPES OF DIGITAL OPTICAL MEDIA (DOM)

There are three major types of DOM in use in the computer industry today:

1. Read-only optical disks, including:
    a. CD-ROM (Compact Disc-Read Only Memory)
    b. Videodisc (Compact Disc Video (CDV), or laser disks)
2. WORM (Write Once, Read Many) disks
3. M-O (Magneto-Optical) or E-O (Erasable Optical) disks

---

[3] The error detection and correction techniques available today can reduce the chances of a system-introduced error to virtually zero; for example, the error detection techniques used in the Data Defense Network (DDN) provide an expected undetected error rate of $4.2 * 10^{-18}$, meaning that if a user sent a full message packet (8,063 bits) over the network every second of every hour of every day, a bit error would slip through undetected only once every one million years! (Taken from p.7 of a 1984 Defense Communications Agency brochure entitled, "DDN Defense Data Network".)

This is not meant to be an exhaustive list of DOM. For instance, read-only optical disks come in a wide variety of sizes besides the 4.7 inch CD-ROM and the 12 inch laser disk, including diameters of 5.25, 8, and 14 inches. However, the types listed above represent the major areas of DOM research, development and marketing today. Each of these forms of DOM has specific advantages and disadvantages; each is appropriate for specific and differing applications.

1. Read-only optical disks

Read-only optical disks are characterized by their permanence of data. Their forte is information dissemination. "The whole purpose of [read-only DOM] is mass distribution. Like books, magazines, microfiche, CD-Audio discs, LP records, and other mass distribution products, the actual...discs are produced in quantity at a replication facility." ([Ref. 19], p.34) While other types of DOM are just as suitable -- and in many applications, more suitable -- for mass data capturing and storage, they are not as useful as read-only DOM for the *publishing* of digital information.

a. CD-ROM (Compact Disc-Read Only Memory)

CD-ROM is indisputably the most common format of DOM in use today. It has been successful in large part due to the early acceptance of standards by almost all CD-ROM manufacturers. But, perhaps as much as standards, size has played a major role in this optical disk format's success. At just 4.7 inches (120mm) in diameter, this small size, together with light weight and durable construction, make for convenient handling, storage, and transportation. This portability translates directly into lowered costs for almost all phases of CD-ROM creation and use.

These factors imply that CD-ROM is superior at least to paper and floppy disks for mass storage and dissemination of textual data, and give strong reasons for its steady growth in such application areas. But CD-ROM is becoming the medium of choice for multimedia applications as well. In fact, there are two competing standards for CD-ROM multimedia applications: DVI (Digital Video Interactive), and CDI (Compact Disc Interactive).

> With a CD's vast digital data storage capacity, multimedia on a personal computer [is] now within reach. This is indeed the idea behind CD-I and DVI....Combine the high-fidelity sound of compact discs, add still images and/or moving pictures, supercharge it with microprocessor horsepower, and the result is an interactive multimedia library. ([Ref. 8], p.28)

DVI (Digital Video Interactive) technology involves compressing multimedia data onto a CD-ROM, allowing up to an hour of video, audio, graphics and other information to be stored on a single disk (without such compression, a CD-ROM could hold only 30 seconds worth of digital video). To play back the images on a PC (personal computer), the CD-ROM player reads the data, sends them to the PC, and the DVI software decompresses them, sending over 30 images to the screen per second. ([Ref. 18], p.14; and, [Ref. 20], p.16)

CDI (Compact Disc Interactive) has similar data compression success, but uses the Motorola 68000 microprocessor inside the self-contained CDI player to decompress the data. The CDI player, which comes with its own remote-control and joystick, can be plugged directly into a standard television and/or stereo equipment, in addition to a computer monitor, to provide the interactive presentation of video, audio, text and other data. ([Ref. 18], p.13; and, [Ref. 20], p.16)

b. Videodiscs

With the exception of its larger size (12 inches in diameter, vs. the 4.7 inch CD-ROM), the videodisc is similar to CD-ROM in most respects. Ironically, it is the videodisc's larger size that is both a major advantage and a major disadvantage vis-a-vis its smaller counterpart. While the larger size detracts from the laserdisc's handling convenience, it does provide a significantly storage capacity: an optical videodisc can hold up to 54,000 images, with an access time as fast as one to two seconds for a single frame. ([Ref. 21], p.103; and, [Ref. 12], p.11) Despite CD-ROM's popularity, the videodisc will maintain its niche in the optical storage world, a position that could well strengthen as more and more applications demand storage capacities beyond those that can be reasonably met using CD-ROM.

2. WORM (Write Once, Read Many) Disk

WORM disks fill another special niche in the world of DOM. They are best utilized for applications which receive new data on a routine, periodic basis, but which require the old data to be retained, rather than over-written.

> [On a videodisc], each document, record or image has an address for use in later retrieval of the data. Although the recording of data is permanent, the address of a former document may be updated, directing the retrieval program to locate a newer, more recent version. ([Ref. 19], p.34)

Like other optical disks, a WORM disk is made of multi-layered materials. To write on a WORM disk, a high-intensity laser is used to heat and permanently alter the information-holding layer, so as to change the way it reflects the light from the low-powered laser used to read the disk. This recording is done on the disk by the WORM drive as needed, until eventually, when the WORM

disk is filled, a new, blank disk is inserted so the continual recording of data can resume. ([Ref. 16], p. 31)

3. M-O (Magneto-Optical) or E-O (Erasable Optical) Disk

Digital data on erasable optical disks can be written and erased, updated and modified, retrieved and manipulated in a similar fashion to other writable storage devices, such as hard disks, Bernoulli cartridges and floppy diskettes. The disks are made of multi-layered materials arranged horizontally in thin wafers. The outer layers are hard and transparent, much like the lacquer on other optical disks. The middle layer, however, contains particles which can be magnetically oriented up or down, but only when the medium is heated by a strong laser.

To write on the disk, the laser heats specific points of the inside medium, and the drive then magnetically re-orients the particles in those specific points. The center layer then hardens as it cools back down, maintaining the particles in their altered orientations. When reading from the disk, these particles reflect the light from a low-powered laser according to their orientation (up or down), in much the same way as the pits and lands of a read-only optical disk. ([Ref. 16], p.32)

M-O disks are most attractive for applications in which the data are somewhat static. In such applications, the information being stored needs to be changed and updated intermittently, but, in between updates, it needs the security and storage capacities offered most cost-effectively by DOM.

IV. OPERATIONAL APPLICATION

A. BACKGROUND

A primary goal of this thesis research was to determine whether optical laser technology and hypertext/hypermedia techniques can be combined and applied in a reasonable manner (logically, technically, financially) to a specific operational application. The context of an operational application was used in order to maintain a central focus for the study. That operational application is the training and readiness of U.S. Navy personnel in the areas of threat recognition and geographic familiarity.

The training of U.S. Navy personnel in the areas of threat recognition and geographic familiarity is continually in need of enhancement. A host of Navy jobs require a sound knowledge of threat units and their capabilities. This knowledge must be constantly refreshed, in both visual recognition and mental, factual recall dimensions. This knowledge must also include familiarity with friendly and neutral platforms, in order to separate them from true threats.

Personnel most in need of such training include Navy flight crews, Combat Information Center watchstanders, ships' lookouts (including "Snoopy Teams"), Bridge teams, and Intelligence analysts. These personnel must be able to instantly recognize threat platforms from a glimpse, seen from almost any angle, and often under conditions of poor visibility and at great distance. Not only must visual recognition be immediate, but so must be the recall of pertinent facts about capabilities such as operating speeds and ranges, weapons carried and their

effective ranges, and the like. The timeliness and accuracy of such recall could potentially, in a hostile scenario, be a matter of life and death, of victory or defeat.

In a similar manner, almost all of these same personnel require familiarity with the geographical operating environment. A knowledge of relative locations of open seas, islands, chokepoints, contiguous nations and their territorial waters, operational bases and airfields, restricted areas, standard operating areas, and such is crucial to safe, effective and successful mission planning and accomplishment.

In addition to training in these areas, there exists the need for a single, effective, quick-reference source to which these personnel can go in operational situations to get timely geographic and threat information. This reference source must be readily available, easy to use, readable, responsive, accurate, up to date, and flexible enough to answer users' queries in the format or formats of their choice. It must accept, tie together, and then cohesively present information taken from a wide variety of sources, and information occurring in a variety of media, including textual, visual, and aural.

B. HOW IS IT CURRENTLY DONE ?

Geographic and threat information is currently found in a plethora of disjoint sources. Unwieldy paper publications, maps and charts, photographs and drawings, 35mm slides and overhead projector viewgraphs, projectors and videotape players and the like are the primary reference and training tools in use today. These resources are largely manual in nature, often leading to laboriously slow, inefficient information retrieval and manipulation. In addition, they are often not centralized, requiring users to go to various locations to have access to

the proper equipment, and to physically collate and coordinate the tools and information for use.

Due to the wide variety and availability of current resources, there is no standardization, often resulting in incomplete training, "holes" in information reliability, and varying degrees of readiness among user organizations. For example, if the first unit of a new class of submarine becomes operational in the naval forces of a particular nation, an organization must update each of their reference publications that include submarines individually. Frequently, there is a waiting period of months before the change is provided as a formal change-issue from the publishers. In addition, new slides and photographs of this class of submarine need to be produced, normally drawn from sources such as commercial defense-related magazines and publications, with the organization able to use only those sources that are immediately available to them (which are extremely limited, for instance, when out to sea).

Currently, the training process occurs most often in the form of briefings, lectures, slide presentations, and training sessions led by Intelligence Officers, Specialists, and collateral duty sub-specialists. Due to the nature of the material and the required knowledge levels, these training sessions can become tautological and mechanical, drilling the attendees over and over until the material becomes ingrained. In other circumstances, the material can be rushed, or presented at a "fire-hose" pace which only a few of the trainees can maintain; the rest retain only as much as they can "swallow", and the bulk of the information flow washes ineffectually away. Because of the current limitations on information presentation,

the trainees receive training which is not individually-tailored, which cannot be self-paced, and which can sometimes be repetitious, inconsistent and/or ineffective.

In addition, this training requires many man-hours to prepare and conduct. The instructors must often divert their efforts away from their primary duties, such as intelligence analysis and reporting. For instance, almost all Naval aviation squadrons are assigned at least one Intelligence Officer. Some squadrons, especially tactical aviation (TACAIR) squadrons such as Helicopter Antisubmarine (HS) and Light Attack (VA) squadrons, need such an officer for very little other than to teach threat recognition to the flight crews. While this is no insignificant task, it is not a full-time job; consequently, the intelligence officer is not used to his full potential in his intelligence role, and is often burdened with a myriad of collateral billets which hinder his participation in and contribution to Carrier Intelligence Center (CVIC) and related activities.

C. HOW SHOULD IT BE DONE ?

It is hereby proposed that the combining of several existing technologies into an interactive computer-based system could meet these aforementioned shortcomings. One system could act as both an integrated, multimedia training tool, and as a centralized, easy-to-use geographic and threat information reference tool. For simplicity, this system will be called the Geography and Threat Recognition (GEOTREC) Training and Reference Tool, or GEOTREC system for short, throughout the remainder of this paper.

Consider first an ideal GEOTREC system, which would be developed if time and money were not major factors, and if existing/developing technologies were

easy to mix and match. Such a system would have capabilities similar to those described in the following paragraphs.

1. Major Features:

The major features of the GEOTREC system would include, but would not necessarily be limited to:

- Viewing (still photographs, three-dimensional line drawings, annotations, overlays)
- Geographical Plotting (maps, charts, overlays)
- Unit Identification (in a Query/Response format)
- Training (point-and-ask, random-viewing quiz, score-keeping)
- Gateway to a relational, multimedia database

Each of these features will be described in more detail in the paragraphs below.

2. Outputs

Although it can seem somewhat backwards and counter-intuitive, it is often most helpful to describe a computer system by detailing its *outputs* before examining its *inputs*. The ideal GEOTREC system would provide multimedia outputs of at least the following types:

- textual, descriptive information relating to threat and friendly platform characteristics, capabilities, order of battle, background information, historical trends, and such;
- digitized maps and charts, in a format which allows for user-selectable scaling and zooming (i.e., the user can point to and mark a spot on the presently displayed chart, and then zoom in or out with that spot becoming the display's new center point), and also allows for user selection of chart type (e.g., road map, Tactical Pilotage Chart (TPC), ocean floor topography chart, map showing political boundaries, or whatever);

- digitized photographs and rotatable three-dimensional line drawings of threat and friendly platforms (ships, submarines, aircraft), weapons systems (missiles, launchers, bombs, guns), sensors (radar, sonar, Electronic Support Measures [ESM] and Electronic Countermeasures [ECM] equipment), and the like;

- overlays consisting of user-generated annotations, highlightings, hypothetical scenarios, and other such markings, to be used, for instance, for briefings, reports, comparison and historical analysis; these annotations could be overlaid by the user on any of the digitized maps, charts, photographs, or drawings;

- motion video, showing actual footage of such things as threat and friendly units underway/inflight, or even panoramic views of approaches to certain landmasses or coastal areas taken from various angles and/or distances; the video would be capable of being stopped by the user at any frame for viewing as a still image;

- an interactive knowledge base of threat unit features and characteristics which could be systematically queried/answered, in order to, for example, match a real-world contact that was sighted but could not be positively identified with a unit stored in the threat database (e.g.: the system would prompt, "Did the observed unit have a two-barrel gun turret?"; if responded to affirmatively, the next question might be, "Was the gun turret located forward, aft or amidships?"; a response of "aft and amidships" would cause the system to retrieve and display, for user scrutiny, pictures of only those units which have two-barrel gun turrets in those locations; and so on).

3. Inputs

The GEOTREC system would receive two major types of inputs: interactive inputs which users would enter on a routine basis, and baseline inputs entered by the system administrator on an as-required basis. Interactive inputs include such information as user name and user identification number for system logon, and responses to system prompts, indicating which commands and selections are to be invoked.

Baseline inputs are those which update and add to the existing information base. They include: new maps, images or video clips added to the system's repertoire, either via receipt of new CDs or videodiscs, or via an image

scanner, scanning images onto hard disk or WORM disk; maintenance of links between related nodes of information; corrections and additions to the textual portions of the system, such as the background and historical data, specifications and statistics, etc.; and updating of the tables in the relational database. The system administrator would be responsible for adding these inputs to the system in a timely manner, while verifying their accuracy, integrity and proper classification to the best of his ability.

Baseline inputs such as CDs and videodiscs would be created and distributed by the national intelligence community and other agencies. Such sources would include the Defense Mapping Agency (DMA) which currently produces a wide range of digital optical media products, the Defense Intelligence Agency (DIA), the Naval Technical Intelligence Center (NTIC), and the two Fleet Intelligence Centers (FICPAC and FICEURLANT).

4. Hardware and Software

The best type of CPU, monitor, and other primary computer hardware on which to base such a system is uncertain. At the very least, the hardware should be fast (e.g., 25Mhz clock speed), have a large amount of active memory (RAM), and a large-capacity Direct Access Storage Device (DASD) such as a 100Mb hard disk. A workstation with a high-resolution monitor, or perhaps even a powerful MS-DOS or MacIntosh system would be needed.

In addition, a fairly explicit suite of peripherals would be required to support the multimedia functionality. These would include at least:

- videodisc player
- CD-ROM player

- image scanner/digitizer
- WORM drive
- full-color graphics capability (e.g., VGA card/monitor)
- laser printer

Beyond the operating system and other standard system software, the required software for such a system would include, at a minimum:

- Hypertext orchestrator
- Special device drivers for the various peripherals
- Relational DBMS (Database Management System) package
- Gateways (to DBMS, other application software/systems such as FULCRUM, etc.)

5. User Interface Scenarios

For what purposes and in what ways would users interact with the GEOTREC system? The system could well be used for many things, and to answer many questions. The following are the major functions that can be presently envisioned.

a. Self-paced Training

The ideal GEOTREC system would allow each user to conduct training at the pace and level of detail most suited to his individual needs. During such a training session, the user could:

- study individual geographical regions or threat units by viewing digitized maps, photographs or three-dimensional line drawings stored on videodisc, CD-ROM or WORM disks; these images would be fully annotated, labeling major land formations, national boundaries, etc., or visible weapons, sensors, and mnemonic features (annotations could be toggled on/off for non-cluttered

viewing); they would also be zoomable, to allow the user to look more closely at a specific feature;

- watch video clips stored on videodiscs or CD-ROM which show actual footage of units maneuvering, submerging, taxiing, and the like, or what the coastal areas and sea lanes look like from the bridge when passing through a specific strait; in addition, the user could stop the video on any frame for viewing as a still image;

- point the cursor to a specific feature of the map or unit being viewed (the unit image could be either a photo, line drawing or "frozen" video frame), click the mouse button, and receive a window-full of textual information about that feature, and/or have the system show a specific, more detailed image of just that feature;

- probe deeper into the information on that unit or one of its features, to view specifications and order of battle information, or read background information, such as how that specific class of unit fits into the broader naval strategy of the nation that developed it, or historical analysis about the past operations of such units, standard operating areas, etc.;

- ask the system to randomly call up non-annotated digitized charts or photographs of units for self-quizzing purposes, with the system asking for responses (e.g., an arrow points to a specific radar on a ship and the user is asked to identify the radar type -- upon a correct response, the system moves the arrow to a specific missile launcher and asks again); the system would provide correct answers when given a wrong response, and could even tabulate a score based on responses.

b. Quick-reference

The GEOTREC system would also provide a single-source, quick-reference system to quickly answer questions arising in an operational setting. Some example scenarios follow to provide a clearer sense of how this might work.

(1) Intelligence Analysis Scenario. An Intelligence analyst receives a report that a source has spotted a patrol boat passing through a particular chokepoint. The patrol craft was reported to have coffin-like missile launchers aft, and was flying an ensign composed of mostly gold and red colors.

The analyst first uses GEOTREC's geographic features to find the chokepoint, and to see if the unit is entering an area of operations of concern to friendly forces. He simply enters the latitude and longitude given in the report, enters the width (in nautical miles) he would like the on-screen display to cover, and receives a display of that area of the world, centered on the given lat/long. From this, he determines that the patrol boat is en route an area where friendly forces are currently located.

The analyst then selects GEOTREC's gateway to the relational database. Using the database management system (DBMS) that GEOTREC has linked to, he queries the database for all naval ensigns of area nations which have at least gold and red as colors. GEOTREC responds with only one possible ensign, pinpointing to which nation the patrol boat belongs.

The analyst then uses GEOTREC's textual information features to get a listing of all naval units operated by that nation. From this, he narrows down the list of possible patrol boat classes to three. Then the analyst clicks on each of those three particular patrol boats, one by one, to view an image which he compares to the source's description of the missile launchers until a match is made. The analyst can then report the contact with increased certainty, allowing appropriate action to be taken by friendly assets in that operating area according to the degree of the known threat posed by the now-identified unit.

(2) Tactical Action Officer (TAO) Scenario. A Tactical Action Officer (TAO) in a ship's Combat Information Center (CIC) receives a message which indicates that aircraft of a particular type are airborne, and heading toward

the ship. Using GEOTREC's unit viewing function, the TAO selects from a menu (or types in at the command line) the class of aircraft mentioned in the message.

The system quickly displays a photograph of that class of aircraft, with menu options to view specifications (such as maximum speed of the aircraft, inflight refueling capabilities, etc.). The TAO clicks on the missile shown being carried under the aircraft's wing in order to receive more information about its capabilities (e.g., maximum effective range). The TAO now has on-screen and at his fingertips the information necessary to make decisions relating to the ship's defensive posture and other such actions.

(3) Aircrew Debriefing Scenario. An aircrew, returning from a Surface Search and Surveillance Coordination (SSSC) mission, heads for the aircraft carrier's Intelligence Center (CVIC) to debrief. They report to the debriefing officer that they saw a large, unfamiliar naval combatant, but due to poor visibility and nearby restricted airspace they could not get closer to take a photograph for later positive identification.

The debriefer accesses GEOTREC's unit identification function to assist in the aircrew's recall. The menu-driven Query/Response session goes as follows:

- GEOTREC: What type of unit would you like to identify? (aircraft, aircraft carrier, surface combatant, submarine, auxiliary, merchant ship, other)
- Aircrew/Debriefer: Surface combatant.
- GEOTREC: What size surface combatant? (major, minor, patrol boat)
- Aircrew/Debriefer: Major.

- GEOTREC: Which of these major features are you certain the unit possessed? (missile launchers, guns, stacks, helipad, torpedo tubes, radar, ALL)

- Aircrew/Debriefer: Missile launchers, helipad.

- GEOTREC: Let's start with missile launchers. Where were the launchers located on the unit? (forward, amidships, aft, ALL)

- Aircrew/Debriefer: ALL.

- GEOTREC: What type of launchers were the forward launchers? (single-rail, double-rail, tube/canister, below-deck/vertical launch, UNSURE)

- Aircrew/Debriefer: Double-rail.

- GEOTREC: What type of launchers were the amidships launchers? (single-rail, double-rail, tube/canister, below-deck/vertical launch, UNSURE)

- Aircrew/Debriefer: Tube/canister.

- GEOTREC: What was the configuration of the tubes/canisters? (single, two side-by-side, two stacked, three side-by-side, cluster of three, four side-by-side, cluster of four, UNSURE)

- Aircrew/Debriefer: Cluster of four.

- GEOTREC: What type of launchers were the aft launchers? (single-rail, double-rail, tube/canister, below-deck/vertical launch, UNSURE)

- Aircrew/Debriefer: UNSURE.

- GEOTREC: Now let's consider the helipad. Where was it located? (amidships, aft)

- Aircrew/Debriefer: Aft.

- GEOTREC: Where was the helipad in relation to the unit's general deckline? (raised, flush, lowered, UNSURE)

- Aircrew/Debriefer: Raised.

- GEOTREC: Was there an associated helo hangar? (yes, no, UNSURE)

- Aircrew/Debriefer: UNSURE.

- GEOTREC: Could the identified unit have been a KRESTA II CG, one of which is pictured here?

At that point, the GEOTREC system displays a photograph of a Soviet KRESTA II class cruiser, as it is the only candidate in the database which met the criteria delineated in the Query/Response session, even with the "UNSURE" responses. The aircrew then scrutinizes the on-screen photograph, has the Debriefer ask GEOTREC to display other shots of the unit to view it from different angles and in different lighting, and makes their determination: the unit they saw was probably a KRESTA II CG.

To confirm this, the Debriefer uses GEOTREC's textual information features to get an order-of-battle listing of Soviet cruisers, indicating where their homebases are. He sees that KRESTA II CGs are indeed based in this theater. The analyst then selects background information, to refresh his memory on where these cruisers normally operate, finding that they often anchor in the protected waters under the same restricted, territorial airspace the aircrew had to avoid. From all this evidence, the analyst corroborates the aircrew's evaluation that the unit they sighted was indeed a Soviet KRESTA II CG.

c. Briefing

GEOTREC could also be used to prepare and/or conduct briefings and presentations. As a supplement to individual, self-paced training, a projector could be used to conduct group training similar to that which is the mainstay of current threat recognition training.

Additionally, presentation slides could be generated using the laser printer. This would allow the briefer to create slides consisting of his own set of annotations and highlights, overlaid on a chart or digitized image. Using these

features, briefings could be created covering topics such as exercise scenarios, planning for actual operations, contingency target exercises and plans, and the like.

D. CAN IT BE DONE THAT WAY TODAY?

The GEOTREC system described above represents an ideal. Such a system, with all its features, is not outside the capabilities of today's technology. But there is one major drawback: the software for building such hypermedia applications is still in its infancy, and must continue to expand and mature to be able to easily and seamlessly handle the various media. Just as important, such software must provide authoring tools that are intuitive and well-documented enough to be easy to use by non-programmers, while being powerful and flexible enough to allow for the broadest possible range of applications to be developed.

A major objective of this research was to investigate the state of the art of such hypermedia application software, using the ideal GEOTREC system described in this chapter as a framework and reference point. The specific goals of this research were as follows:

- to review the areas of hypertext and digital optical media;
- to outline an ideal operational application in which they could be used;
- to attempt to author such an application using an actual hypermedia application development software package;
- to record the difficulties associated with this authoring attempt;
- to draw conclusions and base recommendations on this research as to the relevance, utility and feasibility of developing such an application for actual use in the U.S. Navy.

The author's hypermedia application authoring attempt, with its associated problems and findings, is described in the following chapter (Chapter 5), while the conclusions and recommendations are made in Chapter 6.

V. PROBLEMS/FINDINGS

A. BACKGROUND

In order to direct the study and demonstration of hypermedia features, and to investigate an example of the present state-of-the-hypermedia-art in terms of available software authoring tools, this thesis focused on a specific software package for a specific application. The application: the GEOTREC (GEOgraphic and Threat RECognition) Training and Reference Tool (outlined in Chapter IV); the software: GECI International's *Hyperdoc*, version 1.12.

Hyperdoc was chosen for several reasons; first, because it is MS-DOS based. Apple Computer and other developers of Mac/OS based software have taken the lead in developing hypermedia authoring tools and applications. This is largely due to the graphical user interface (GUI) inherent to and so familiar in the Apple/Mac computer world. Hypermedia packages such as HyperCard (which Apple Computer now bundles with almost any purchase of Macintosh hardware), SuperCard and Guide have become widely used for hypermedia applications.

The problem remains, however, that the de facto standard PC (personal computer) operating system in the U.S. Navy is MS-DOS. *Government Computer News* estimates that, by January 1990, the Department of Defense had purchased up to 600,000 IBM-compatible (i.e., MS-DOS compatible) Zenith PCs. ([Ref. 22], pp.1,72) In light of this, it was considered important for this thesis research to utilize an MS-DOS based hypermedia software package.

In addition, Hyperdoc was chosen for its advertised capabilities and flexibility. Advertisement literature such as a booklet published in 1988 entitled simply "Hyperdoc", claims: "HYPERDOC has a CD ROM [sic.] processing interface, which allows the use of the best features of this new medium facilitating processing and storage of a very large volume of documents in reduced format"; and, "Since the document recording format is totally independent of all operating systems and all display peripherals, HYPERDOC documents can be run on any computer, from mainframe to PC. this [sic.] degree of adaptability guarantees its long life despite equipment advancement, a quality essential for the maintenance of an expensive documentary archive base on a computer system which is constantly changing." These highly desirable characteristics of hypermedia software will no doubt be integral to future versions of Hyperdoc as the product matures. Unfortunately, the only version of the software available for this research, Hyperdoc 1.12, did not incorporate either one.

B. PREPARATION

In order to begin prototyping the GEOTREC application using Hyperdoc, several preparations had to be made, the first being computer hardware. The Naval Ocean Systems Center (NOSC) in San Diego, California, greatly assisted this research effort by loaning a FULCRUM system, consisting of a Compaq Deskpro 386 AT (with a 60 Megabyte internal hard drive, mouse, and VGA graphics), a Pioneer LD-V6000A Laserdisc player. and a 12-inch Sony color monitor, in addition to the FULCRUM software. This hardware was placed in the Naval Postgraduate School's Optical Laboratory. in order to be able to utilize other

hardware available in the Lab as needed, such as CD-ROM drives, an optical scanner, and a laser printer.

The second preparation was to obtain or produce digitized images around which the prototype could be built. Ideally, these would be high resolution images of aircraft, ships, submarines, weapons systems, naval installations, airfields and the like (whether digitized photographs, line drawings or diagrams, maps or charts, or even "artist conception" drawings), stored on optical media such as CD-ROM or laserdisc. NOSC's FULCRUM system came with one laserdisc of maps ranging from a world map to detailed charts of the Korean peninsula, ideal for demonstrating the capabilities of this prototype.

Finding a suitable CD-ROM, however, was unsuccessful. The only example of a CD-ROM appropriate to this research was seen advertised in the October 1989 issue of *CD-ROM EndUser* magazine (p.79). It is a CD-ROM with the title, "Dick's Some Of The Earth's Planes" (a tongue-in-cheek name which plays on the title of the authoritative Jane's All The World's Aircraft encyclopedic series). The reader is told that the disk "...contains hundreds of black-and-white and color aircraft images", in addition to silhouette images (in .PCX format), and information on each plane such as "...air speed, length, wing span, crew, service, range and mission". Again, such a disk would be ideal for the aircraft portion of the GEOTREC prototype's database. Unfortunately, the version of Hyperdoc used for authoring this prototype was not capable of controlling either a CD-ROM player, nor FULCRUM's laserdisc player, so the search for available optical disks became inconsequential.

For this reason, efforts were directed at scanning images onto hard disk. These images were obtained from such sources as issues of *Jane's Defence Weekly* and *U.S. Naval Institute Proceedings* magazines, as well as Norman Polmar's comprehensive Guide To The Soviet Navy. [Ref. 23] The process was difficult and time-consuming for several reasons, the most significant of which was Hyperdoc 1.12 only accepts images that are stored in the .PCX standard file format. Because of the limited nature of this file format, image selection had to be very restricted; for instance:

- only photographs with sharp object/background contrast could be utilized (for example, photos of submarines were particularly difficult to use, as they almost always consist of a very dark object, the submarine, against a dark background, the surrounding ocean);
- only small photos (i.e., something less than 4 x 5 inches) could be utilized, as anything larger would translate into too much digital information to be stored in the .PCX format; larger photos could be scanned in and then reduced in size to meet the .PCX limitations, but at a great loss of detail;
- the images obtained were at very low resolutions (i.e., normally 75 dpi [dots per inch], with some at 120 dpi, depending on the size of the original image and the scanner used), again, losing all but the most obvious details.

Two scanners were used to capture the images: the Xerox 7650 Pro Imager, and the Discover 7320. Although the large-screen Xerox 7650 System had by far the more capable scanner (including exceptional image viewing, editing, and manipulating capabilities), the system often crashed during conversion attempts, as its RES (Xerox's proprietary image file format) to PCX conversion utility was not formally supported by the resident software release. Each system crash would require a twelve minute reboot process. In addition, the Xerox was capable of handling only black-and-white images. The Discover 7320 was more robust and

easier to use, but provided no way to preview the image before scanning, or to view or manipulate the image after scanning, so all adjustments had to wait until the image was imported into ZSoft Corporation's PCPaintbrush (the use of which is further explained in the following paragraphs).

Adding to these difficulties, Hyperdoc's conversion tool can only handle a small subset of the several .PCX variants within the standard. That is, only .PCX files that are output from a paintbrush program, such as Zsoft Publisher's Paintbrush or ZSoft PCPaintbrush can be correctly converted by Hyperdoc. Consequently, a copy of ZSoft PCPaintbrush had to be obtained, loaded, and learned. All .PCX files created by the scanning process had to be loaded into PCPaintbrush (even if no editing changes had to be made), and then saved back to disk. Only then could the Hyperdoc conversion utility be used to successfully convert the .PCX file to Hyperdoc's proprietary file format (.YG0). If the scanned .PCX files were directly converted by Hyperdoc without first calling them into PCPaintbrush, the final product came out distorted, with a severely "squashed" appearance (i.e., full width, but only one-third height).

C. EVALUATION OF HYPERDOC

Once the images had been scanned in and converted to Hyperdoc's .YG0 format, the authoring began. The authoring, in and of itself, proved to be quite a challenge.

It should be noted up front that Hyperdoc does have some good points. For instance, the Hyperdoc, USA, office in San Jose, California, is staffed with friendly, accommodating personnel, who were always available via telephone for technical support and advice. Also, Hyperdoc's file extension scheme was

intuitive, consistent and helpful, distinguishing each of the many types of files used by Hyperdoc, and yet keeping them appropriately related (with one minor exception, noted later in this chapter). In addition, the windowing features were substantial, allowing for several helpful features, such as the automatically added "move menu" option appended to all menus created with Hyperdoc's "MENU" command. Most importantly, Hyperdoc is moving in the right direction for supporting hypermedia application development, with a wide variety of good ideas and advertised features which, if brought to fruition in later Hyperdoc versions, will comprise an excellent hypermedia package.

On the negative side for version 1.12, however, the list is long. In general, the difficulties encountered by the author were primarily in the areas of:

- poor documentation (both on-screen and paper);
- non-intuitive, inconsistent user interface;
- many capabilities were more limited than advertised;
- non-robust, incomplete, non-seamless, unpolished package;

The paragraphs below provide a sampling of specific Hyperdoc "rough spots".

1. Poor Documentation

The documentation that accompanied the Hyperdoc software was sparse, skeletal, and incomplete. For example, the written documentation included no index for quick-reference, and no printed sample *action files* (Hyperdoc's term for program files written in Hyperdoc's command programming language), although

simplistic action files from a sample application were included on diskette. Other specific examples of problems in the documentation are as follows:

- The documentation contained a plethora of misspellings, missing words, incorrect grammar, run-on sentences, and other such elementary errors; here is a sampling:
  -> p.43 of Hyperdoc Command Reference, under "MENU" entry, usage is spelled "USUAGE", and the very next sentence reads, "...Returns 0 if user presses ape key."
  -> pp.27,28 of Hyperdoc Command Reference, the entries are out of alphabetical order ("DISDOC", "DISTEXT" and "DISTEXTSTOP" are listed after "DOS" and "DOSEXECUTE")
  -> p.25 of Hyperdoc User's Guide, under PAN & HOME entry, reads:
  "The Home function is use to re-align the image with the upper left corner of the document with the upper left corner of the screen."
  -> p.29 of Hyperdoc User's Guide, under PASTE entry, reads:
  "The Paste function is used to merge a different document into the one currently being editing on."
  -> p.31 of Hyperdoc User's Guide, second paragraph on the page, reads:
  "In order to move a document created at higher scales (smaller images) into a lower one (larger images) the Convert scale function is used. The effect of this function is the same as if the document were created at the lower scale factor to."

- In the Hyperdoc User's Guide (p.53), the instructions on how to use the Library editor to create new commands is severely lacking. For example, in describing the menu choice "EDIT", the instructions read only, "This option enables the developer to: 1. modify an existing function; 2. create a new function" -- and nothing further.

- In the Hyperdoc Command Reference, the "CLOSEWIN" command is particularly difficult to find, as it is neither included in the table-of-contents listing, nor indented and **bolded** for emphasis as are all the other commands; furthermore:
  -> the reference lists this command as "CLOSEWINDOW",
  when in reality it is "CLOSEWIN";
  -> it erroneously states that this command can only be
  used after the "OPENWND" command is issued (in fact, "CLOSEWIN" *must* be used in conjunction with several other commands, such as "DISTEXTWIN");
  -> its reference to the "OPENWND" command just described
  is incorrect, as the actual command is "OPENWIN".

- The documentation does not provide a listing of available fonts by file name (e.g., "System72.fnt"), but offers no indication of what these available fonts would look like (pitch, type, etc.) when invoked by an action file.

- In several places (e.g., in the Graphics editor), some of the menu selection names given in the documentation differ from those names actually given on-screen; case in point, the SPECIAL TOOL named "HDSCALE" is listed in the Hyperdoc User's Guide as such, but appears on the menu as "HDECHEL" (which is no doubt the abbreviated French equivalent for "HDSCALE", as Hyperdoc was originally written in French); also, one of the tools is listed as "CONFIG.EXE" in the User's Manual, but appears as "HDCONFIG" on the menu.

- Very little explanation or instruction is provided in the documentation for using Hyperdoc's SPECIAL TOOLS, either in the paper documentation (where there is a total of one half page given to the entire topic) or on-line (where there is no assistance whatsoever).

2. Non-intuitive, Inconsistent User Interface

The user interface provided by Hyperdoc 1.12 was largely non-intuitive and inconsistent, making for a very steep and long user learning curve. Menu options were often confusing and nondescript. In some cases, the same key was assigned different functions at different times for no apparent reason. While some of the problems were little more than annoyances, others made it difficult to learn and accomplish what was intended in a reasonable amount of time. Some of the specific user interface difficulties were:

- In the Graphics editor, under the FILL POLYGON option, after designating coordinates for the polygon's corners, the user must press ESC, instead of something intuitive like ENTER, to get the editor to close and fill the polygon.

- Also, in the Graphics editor, a RIGHT TRIM option is provided to allow the user to slice off part of the right edge of the graphic if desired (it is perplexing why only RIGHT TRIM is available, and not also LEFT, TOP and BOTTOM TRIM); the problem is, however, that the original graphic is never cleared from the screen, and the new version of the graphic, with its trimmed right edge, is redrawn directly atop the old graphic; consequently, the user is prevented from viewing the new graphic (as is appears with its

right edge trimmed off) prior to confirming whether he wants to save it to disk and replace the old graphic.

- Upon exiting from the graphics editor, the image that was being edited is automatically redrawn at one-fourth scale in the upper left corner of the screen and on top of the original image; not only is there is no apparent reason for this, but in the case of complex images, it can be extremely time consuming, even on a 16Mhz 386 machine; furthermore, following this redraw, a box appears atop the redrawn image with the enigmatic message, "Display drawing at scale required for level 1" <newline> "Hit F1 to modify the display"; that's it -- no further instructions, explanation, or indication of what module or function you are in, or what your other options are; in order to exit this and not modify the display, the ESC key must be used, although the user is not told this in any documentation other than the tutorial.

- In the Descriptives editor, when an object (similar to a function or procedure in structured programming languages) from an action file is called up for editing, none of the instructions that are normally shown in the Object editor are displayed -- the user has to rely on recall as to what keys are used to edit, save, exit, etc.; in addition, the ESC key, which is normally used in the Object editor to produce an END_OBJECT designation, does not produce the same result (instead, although no indication is given on-screen, an END_OBJECT designation is added automatically upon exiting from the Descriptives editor, unlike the Object editor).

- Throughout Hyperdoc, the indication for the user to wait while the system is processing is inconsistent: sometimes it appears as a little hourglass in the center of the screen (good), sometimes as successive dots emanating from the left, center edge of the screen (hard to see, and meaning unclear), and, more often than not, nothing appears at all (undesirable).

- Concerning Hyperdoc menus, the user is forced to highlight the desired selection (using the mouse or up/down arrow keys) and then press ENTER (or the left mouse button); the user cannot simply press the number or the first (or other designated) letter corresponding to the desired menu choice to make a selection; and while it is convenient to be able to use either keyboard or mouse, this can be very tedious for an advanced user who would rather enter series of letters in order to speed through a series of known menus (as provided in LOTUS 1-2-3, inter alia), instead of having to go through every single menu one by one.

- Hyperdoc 1.12 exhibits a bizarre, non-intuitive, inconsistent, and yet ubiquitous, use of the ESC (escape) key:
  -> when attempting to exit from an editor or other feature, the message typically appears: "Hit ESC to finish"; one would think this was functioning as an exit confirmation ("Do you really want to exit this feature?"); however, the user is not allowed at that point to undo his selection (i.e., to say, "No, that's NOT what I wanted to do" by pressing ESC or some other designated

key); thus, hitting ESC to finish is apparently an unnecessary step which quickly becomes a nuisance;
-> in fact, at some points where "Hit ESC to finish" appears, pressing any other key besides ESC (such as ENTER) has detrimental consequences; for instance, in the Object editor, after editing and exiting from an action file, if the user presses ENTER at the "Hit ESC to finish" prompt, he is again asked for the name of a file to be edited, apparently allowing the continued use of the Object editor without having to reselect if from the menu; this would be a convenient feature if it worked; however, the outcome of any subsequent action file editing is unpredictable, can often engender a message that the file being edited (even if small) is too large and needs to be divided up, and may sometimes result in the loss of data;
-> in the drawing menu, under the CHANGE LEVEL option, the user is directed to move the crosshairs to designate the desired center for the new level of the drawing, after which he must press ESC to execute that designation (although this is NOT mentioned anywhere on the screen or in the documentation, except in the tutorial);
-> in the Object editor, as mentioned previously, the ESC key is used to produce the "END_OBJECT" character, instead of something more intuitive and self-explanatory such as the END key (or the CTRL-END key combination, since END is used to move the cursor to the end of a line).

- In general, on-screen instructions (as to the function of various keys, how to get help, how to get to the previous screen, how to exit, etc.), as well as error messages, are cryptic, misspelled, improperly spaced, and woefully inadequate (e.g., in the Graphics editor, under the CONVERT SCALE option, the message to the user reads: "To be completely displayed on the screenthe drawing must be in the scale 4" <newline> "initial scale: +", where the "+" is followed by 1, 2, 3 or 4, and the words "screenthe" are run together as written here).

- The SCALE FACTOR, used in the Graphics editor and other places, is a user-changeable variable which has a possible range of 1 - 4 (for 1/1, 1/2, 1/3, 1/4 scale); the use of words such as "full-size", "half-size", "third-size", and "quarter-size" to represent these quantities would be much more clear; all the more so because the SCALE FACTOR is used hand-in-hand with another such variable, DISPLAY LEVEL, which also ranges from 1 - 4, allowing the two to be easily confused (perhaps DISPLAY LEVELs could be expressed in letters such as, "A, B, C" etc.); furthermore, a SCALE FACTOR of 3 cannot be used other than to temporarily view the image at that scale (images can only be saved at SCALE FACTORs 1,2 and 4).

- Hyperdoc has given non-mnemonic names to its SPECIAL TOOLS utilities: "HDVC" for the conversion interface utility which can only be used with image files scanned in using a Microtek 300A scanner; "HDHYPPB" for the

Hyperdoc-to/from-PCX conversion tool; "HDEDCMM" for the command editor; and so on.

- The Hyperdoc files containing the application author's programming code are sometimes referred to as "action files", and sometimes as "object files" or "objects" (they all end with a ".YOB" extension, "OB" presumably for "OBject"); however, Hyperdoc in reality makes a distinction between objects, which are like functions or procedures in structured programming languages, and the files which contain those objects (each file can be comprised of any number of objects); as such, the action files should be consistently referred to as such (with perhaps a ".YAF" extension), and edited with an "Action File" editor (vs. an Object editor as it is presently called).

- Many of the Object editor's key combinations are confusing and hard to remember; for instance, the CTRL-N combination is used for MOVE BLOCK, when the more intuitive CTRL-M (M for Move) is not being used elsewhere; CTRL-G is required for REPLACE BLOCK; CTRL-R for COPY BLOCK; etc.; in addition, while the F1 key does bring up a help window listing all these key combinations and their usage, this window cannot be left open for reference purposes during editing (every time the user needs to refer to it, he must stop editing, call up the window, look at and remember the key combination, and then exit the window, return to editing, and use the proper key combo).

- In many instances, Hyperdoc will not allow the user to undo undesired entries; for instance, when attempting to editing a graphic document, the user is asked to enter first the x coordinate and then the y coordinate of origin (the screen coordinates at which the upperleft-most corner of the image will begin), and is then asked for the SCALE FACTOR; after the user enters a number for the x or y coordinate and presses ENTER, he cannot go back and change the entries, or get out of that function; pressing the ESC key at this point (the intuitive way to say "No, that's NOT what I want to do -- take me back") is the same as pressing ENTER; thus, the user is forced to either go through with that operation (which could involve another lengthy graphical image drawing) or reboot the entire system (more often than not the more expeditious of the two options).

- The Object editor and the Text editor shared many of the same problems in terms of user interface:
  -> with each and every backspace movement or delete keystroke, the entire screen is redrawn, which is not only slow and annoying, but also prevents tracking the cursor to see how much is being deleted until all the blinking stops;
  -> no clear indication is given as to whether the editor is in the insert or typeover mode, such as changing the cursor's size or color; all that is provided is a small "Rempl." (assumed to be abbreviation of French word for replace) or "Insert" in the upper left corner of screen;
  -> the user must enter hard returns at the end of every line; if inserting

a new line between two adjacent, existing lines, the user must hit return, then type the text of the new line, then hit return, which puts in an unwanted blank line, and then delete the blank line just created; if this is not done, the text of the third line will begin immediately after the last character of the second line when the object file is subsequently retrieved); this is especially agonizing when a large portion of text (multiple lines) is blocked and then moved or copied -- once the block move or copy is complete, a hard return must be inserted after every line, and the blank line which is produced by inserting each hard return must then be deleted one at a time;
-> when text is blocked down to the end of an action file, and that block is deleted, any attempts to scroll back up in the file (i.e., repeatedly moving the cursor upward) results in the cursor disappearing off the top of the screen, and the editors locking up, requiring a system reboot;
-> the editors also lock up when the user attempts to place a copied block of text immediately above the original block, requiring a system reboot.

3. Limited Capabilities

Hyperdoc version 1.12 includes no CD-ROM interface capability, and can only handle .PCX graphics format (with the exception of only one, little-known "Microtek 300A" format scanned files). In addition, each Hyperdoc graphical image file can reportedly provide resolutions of up to 4000 x 6000 pixels -- however with the scanner and .PCX limitations, this capability is essentially unusable.

In addition, while some versions of Hyperdoc can interface with a videodisc player -- albeit, only one specific model of one specific brand (i.e., Pioneer LD-V3500) -- no such capability was included with the software used for this research. Furthermore, action files (i.e., ".YOB" files) can be created in other word processors and then imported as ASCII files into Hyperdoc, but once imported they can *not* be edited using other text editors (QEDIT, WORDPERFECT 5.0, and EDLIN all failed); Hyperdoc's Object editor apparently

changes the commands in the action files to numbers, and then inserts non-printable characters as in a compiled, executable file).

Finally, gateway commands to other application programs (e.g., "WAITEXECUTE") do not work with any but the smallest executable programs (e.g., .COM or .BAT files), due to memory limitations. That is, enough of Hyperdoc stays resident in memory (vs. written to disk) that attempts to use other applications such as INGRES 5.0, PARADOX 3.0, and WORDPERFECT 5.0 were all futile (a tenuous workaround was used in GEOTREC and will be described later in this chapter).

4. Non-robust, Unpolished Presentation

In addition to being accompanied by poor documentation, providing a laborious user interface, and a reduced capability to interface with other products, Hyperdoc version 1.12 failed to present itself as a robust, complete, mature software package. It had difficulty handling user mistakes, unexpected key combinations, out-of-range inputs, and other such anomalies that robust software should be able to manage, even if only by displaying explanatory error messages and returning the user to the state of the program prior to the anomaly's occurrence. Moreover, Hyperdoc did not provide a seamless, polished look and feel the author expected of a mature software package in the $1000 price range.

The following are specific instances that support these judgments:

- Many of the on-screen instructions and error messages are still in French; for instance: when saving action file changes made in the Object editor, the user is asked to enter "Nom de la copie: "; when the number of windows opened in the application at any given time exceeded memory constraints, the error message appears, "Trop des fenetres!..."; in the Graphics editor, when designating the center for the new version of the graphic, the user is prompted with "Nouveau centre:"; and even the executable file which

produces the main menu for Hyperdoc Tools is named HDOUTILS.EXE (as opposed to HDTOOLS.EXE).

- When the main Hyperdoc Tools menu is displayed by HDOUTILS.EXE, the menu is noticeably off-center on the screen, just under its title-caption "Hyperdoc Tools" which *is* centered, thus exacerbating the faulty positioning of the menu.

- When in the Graphics editor, if the user accidentally selects an incorrect menu choice, there is no way to abort the on-going function (e.g., by pressing the ESC key); in the case of a large, complex graphic, this means waiting as long as twenty seconds or more for the graphic to be drawn, then exiting from that accidentally chosen function, returning to the menu, selecting the correct function, and then waiting another twenty seconds for the graphic to be drawn under the correct function.

- Also in the Graphics editor, the FILL POLYGON option does not work properly; the user is first asked to designate the first corner coordinate, then prompted to specify the color of the fill, instead of being asked for more corner coordinate designations; after selecting the color, the user is asked for another corner coordinate, followed immediately by a request to once again specify the color of the fill; finally, after designating all desired corners, the user is directed to press the ESC key (as opposed to the ENTER key, a much more intuitive option at that point) to fill the polygon -- however, the ESC key returns the user to the Graphics editor menu, without the polygon being drawn or filled.

- Again in the Graphics editor, when attempting to exit the EDIT option, the system locks up entirely, requiring a cold system reboot (i.e., no keyboard inputs are accepted whatsoever, not even CTRL-ALT-DEL)

- And finally in the Graphics editor, under the SAVE BLOCK option, the system locks up when the user attempts to overwrite the graphics file currently open for editing; Hyperdoc should either tell the user it is an invalid operation and that he should save the block to a different file name, or else allow him to do the overwrite.

- When finished with image conversion (PCX-to-Hyperdoc), using the "HDHYPPB" special tool invoked from the Hyperdoc Tools menu (HDOUTILS.EXE), the user is not returned to that menu, but rather is taken out of HDOUTILS completely and returned to the DOS prompt with no explanation message en route.

- When Hyperdoc is transitioning between modules, the apparently meaningless, system-level message, "HYPERDOC : final control = 0 / 0" is almost always displayed on an otherwise blank screen; this, in addition to the ubiquitous "Hit ESC to confirm" message described previously, make for very rough, non-seamless transitions between modules and functions.

- At several other transition points (e.g., within the DOCUMENTS option on the main Hyperdoc Tools menu, after entering "G" for document type, and then the document name), the screen totally blanks out, followed by the re-display of the exact same screen, with just a slightly different data-entry window overlaid; retention of the same background with changes to the data-entry window would be preferable.

- The "browser" feature only works after the PROCESSDOC command has been invoked (i.e., only when image/text file is displayed on-screen awaiting user selection with mouse); in itself, the browser is a very useful feature, but it would be much more so if it tracked action file code-generated images or text (e.g., DISDOC, MENU, DISTEXT, etc.); also, when clicking on a browser icon (representing a "PROCESSDOC" graphic or text), or after exiting the browser, the screen is not cleared, so that if the document being processed is not full-screen, the browser can still be seen in the background, providing unnecessary screen clutter.

- Some inconsistencies exist in the object programming commands used to create action files:
  -> most screen-output commands, such as "WRITE", allow the author to select both the background and the text colors (a good feature), whereas some, for instance the "DISTEXT" command, accept only one color parameter (with no corresponding list provided in the documentation as to what two colors result from each single parameter);
  -> the "DISTEXTSTOP" command does not accept a cursor parameter, as indicated in the Command Reference Guide; rather, it displays the number given for that parameter on the screen as if it were part of the text being displayed;
  -> no logical AND or OR capability is provided (e.g., the hypermedia author cannot program the equivalent of: "IF x > 3, OR x = 2, THEN..."); a related limitation is found in the "CHOICE" command (similar to a SWITCH statement in the C language, or a CASE statement in Pascal) in which it is impossible to specify one command for multiple, successive cases (e.g.: cannot say "CASE=1, 2 or 3: <command1>"; instead it must be spelled out, "CASE=1: <command1>; CASE=2: <command1>; CASE=3: <command1>;", creating redundancy, difficulties in program maintenance, etc.)

D. THE PROTOTYPE GEOTREC APPLICATION

The prototype GEOTREC application was designed with a twofold purpose: first, to explore Hyperdoc's hypermedia capabilities in an actual application; and,

second, to produce a prototype which included some of the functionalities essential to an ideal GEOTREC of the kind outlined in Chapter IV.

The design was based on several key assumptions:

1. a login procedure would be required to limit access to authorized personnel only;

2. two access levels would be required, a read-only level for the ordinary authorized user, and a read/write level for a System Administrator who would be responsible for updating,. modifying and maintaining the application (e.g., adding in newly scanned images, or updating the list of valid users and their passwords, and the like);

3. the user should be able to access other applications, such as databases and the FULCRUM application referred to earlier in this chapter, via the GEOTREC/Hyperdoc shell;

4. the majority of information the user would need in order to access and use all features of the system would be provided on-screen; and,

5. the code would be general, consistent and documented enough to permit easy modification and/or adaptation for future uses beyond this thesis.

The appendices accompanying this thesis provide the clearest possible picture of the GEOTREC prototype developed for this thesis research. Appendix A is a collection of GEOTREC screens and accompanying explanatory text, organized into a tutorial-like run-through of GEOTREC's salient features. Appendix B is a copy of the GEOTREC action files (.YOB), containing the programming code written in Hyperdoc's command language, as well as the few DOS-level batch files (.BAT) written in plain ASCII format, which were used in the prototype.

The batch files are included as an integral part of the gateway workaround, needed, as previously mentioned, because Hyperdoc's gateway commands do not function within the MS-DOS 640K memory constraint as required by GEOTREC. To reiterate, a gateway should ideally allow the user to exit the primary

application, enter and normally use a secondary application, and then return to the primary application just where he left off.

To get around gateway difficulties to other applications in Hyperdoc, a very rough, but functionally equivalent route has been taken. Any action file which would normally call a secondary application directly (using Hyperdoc's WAITEXECUTE command), instead calls a DOS-level batch file (by using the DOSEXECUTE command). That batch file then calls the secondary application, remains resident in memory until the application is finished running, and then continues by calling the GEOTREC.BAT batch file, thereby restarting GEOTREC from the beginning (including logging in, et al). While this is clearly unacceptable for a full, completed application, it was considered an appropriate alternative for this working prototype.

The general GEOTREC design consists of three major functional areas:

1. the user-driven viewing and investigation of digitized unit, weapon, and sensor images, and their related explanatory text using hypermedia links;

2. the system-driven Recognition Quizzer, which randomly displays images of units, weapons, sensors, etc., and asks the user to identify the item from a provided list of choices; and,

3. the aforementioned gateways to other applications.

All users initiate interaction with GEOTREC by logging in, using both an alphanumeric user identification code, and then an alphabetic password. After login, the viewing and investigation of images and related information is reached through a series of menus. This menu hierarchy is designed to help the user channel his search by logically breaking the available database of images into

functionally related areas (e.g., fighter aircraft, major surface combatants, naval guns, airborne radar, etc.).

Once a specific image is selected for display, say, a picture of a certain class of surface combatant, Hyperdoc's hypermedia links allow the user to click on any major feature of the unit to receive further information. Generally, that information initially comes in the form of a short label (e.g., "This is an SA-N-3 missile launcher."). Concurrently, an instruction box appears, asking the user to either press "M" for More information about that particular feature, or to press any other key to continue clicking on other features.

If the user presses "M" for More information, either a text box or a more detailed image of the selected feature appears. No matter which one of these new displays is overlaid on the initial image, it also contains hyperlinks, so that clicking on any major object phrase or feature will call more information. Again, this new information display, either image or text, will in turn have hyperlinks for further inquiry.

And, as described in Chapter IV and shown in Appendix A, the "browser" can be invoked at any time during this "click-and-search" session. This allows the user to view a diagrammatic "roadmap" of the information path taken to that point. It also allows him to click on any part of that roadmap, instantly taking him to that node in the search diagram, and displaying the information (text or image) represented by that node.

The Recognition Quizzer puts a different twist on the user's information-gathering and learning experience: instead of the user specifying what he wants to see and then having it displayed, the system displays what it randomly "wants" the

user to see, and then asks the user to indicate what it is he is seeing. The user is given three chances to correctly identify the displayed item, after which, if unsuccessful, he is given the correct answer. Whether he has identified the item accurately or erroneously, the user is permitted to further investigate the image using its hypermedia links in the exact same manner as previously described. When he is finished with such investigative forays, the Quizzer returns to its random generation of images for continued testing and learning.

Finally, GEOTREC's gateways allow for the incorporation of various other applications into what ideally appears to the user as one cohesive, didactic whole. In this prototype, gateways are provided to the following applications:

- FULCRUM, a geographic information and database system developed at NOSC San Diego. Among a wide range of other functions, FULCRUM's capabilities include the retrieval and display of digitized maps and charts which are stored on laserdiscs. FULCRUM allows the user to overlay annotations and data sets (e.g., unit positions, tracks, range circles/ellipses, and much more) on the geographic images, and would be invaluable as a geographic training and reference aid in the context of the GEOTREC application.
- The THREAT DATABASE, an Ingres 5.0 application which provides for the retrieval of requested data (text only), either menu-driven or using SQL (Standard Query Language) queries;
- PARADOX 3.0, an easy-to-use, yet extremely capable relational DBMS (database management system) that could be used for locally generated database requirements;
- HYPERDOC TOOLS, allowing the System Administrator access to the full range of Hyperdoc's tools for application modification and maintenance.

These applications provide capabilities far beyond what could be produced with Hyperdoc's authoring tools; indeed, therein lies the beauty and importance of "other-application gateways" in any hypermedia software package. Moreover,

these in no way comprise the definitive set of such relevant applications, but rather only serve to illustrate the types of applications which can be linked through the GEOTREC application.

Due to time constraints, not every option of every menu in this prototype is presently available. In addition, due to previously described scanner difficulties as well as hard disk constraints, only a limited number of images have actually been incorporated into the prototype. For these reasons, an action file named "GENERIC.YOB" (see Appendix B) was created, and is called when appropriate to indicate to the user the general format in which a response to his request would be displayed once that option or image became available. Obviously, in a complete version of the GEOTREC application this file would be superfluous.

VI. CONCLUSIONS/RECOMMENDATIONS

A. CONCLUSIONS

The surveys of hypermedia and DOM (digital optical media) technologies conducted as background for this thesis (Chapters II and III, respectively), as well as the prototype implementation of the GEOTREC application which formed the focal point of this research (Chapters IV and V, and Appendices A and B), have yielded conclusions and recommendations, the most significant of which are addressed in this final chapter.

Of the major conclusions to be drawn from this study, perhaps the most evident is that Hyperdoc version 1.12 presented limitations which negatively impacted the authoring of the GEOTREC prototype. However, despite version 1.12's numerous drawbacks, Hyperdoc's basic design is sound, with some presently implemented capabilities which are solid, and an advertised future direction which appears well-chosen. Indeed, more mature versions of Hyperdoc, in which shortcomings such as those outlined in Chapter V have been competently addressed and rectified, could well become a front-runner in the field of hypermedia software and application generation.

Additionally, it can be concluded that the MS-DOS operating system environment may somewhat constrain hypermedia applications like GEOTREC. These applications, which involve handling numerous graphics and gateways to other applications, often require large amounts of primary memory, beyond the 640K standard to MS-DOS. And while various expanded memory options are

now readily available for MS-DOS, they can dramatically increase system complexity and expense. More importantly, it can be assumed that very few of the MS-DOS based PCs currently in Fleet use have extended memory capabilities. Furthermore, aside from some of the very latest releases, few of the software packages which would be useful in such an application (such as those used in this thesis research and prototype, including Ingres 5.0, Paradox 3.0, WordPerfect 5.0, and Hyperdoc version 1.12) can utilize expanded MS-DOS memory.

This thesis has underscored advantages of using both hypermedia and DOM for an application like GEOTREC. The process of learning is done much more efficiently and effectively when information is presented at the proper pace in a variety of media. Hypermedia not only provides multimedia stimuli, but also allows user-paced and -directed instruction which accommodates each user's own learning abilities, strengths and weaknesses. In addition to superb educational characteristics, hypermedia provides an ideal avenue for quickly referencing desired information by allowing speedy location and retrieval in multiple formats, with related support data just a mouse-click away.

High-resolution graphical images are enormously expensive in terms of digital storage, particularly if they are in color or a fine-grain grey-scale. Moreover, the images utilized in this application are static in nature, changing only when corresponding updates to the order-of-battles occur in applicable military services. DOM not only provides permanent storage of this information in a secure, convenient, cost-efficient format, but also permits its rapid retrieval, without using up valuable hard disk space needed by application programs.

A final conclusion from this investigation is the utility and sagacity of authoring, that is, providing tools so the end user can develop prototype applications to meet his own needs. Prototyping has become one of the most valuable system design and development tools, particularly for relatively small, non-complex applications such as GEOTREC. Allowing the user to actually perform prototyping through authoring not only ensures that user requirements are being met, but that user acceptance will be forthcoming, leading to overall system implementation success.

B. RECOMMENDATIONS

The primary recommendation of this thesis is that the problem of geographic and threat recognition training and readiness be addressed with hypermedia and DOM solutions. Whether in the format of the GEOTREC prototype developed for this research effort, or in a different genre, it is clear that a meshing of hypermedia and DOM technologies would provide the best solution to a pressing operational need.

In order to make this feasible, the use of high-quality hypermedia authoring software is recommended. Such software must provide clear, concise on-line documentation (preferably implemented with hypertext features), with in-depth coverage provided by accompanying paper documentation. The documentation should include glossaries, indexes, quick-reference sections, a broad variety of sample code files and programs, toll-free/24-hour assistance phone numbers, and the like.

In addition, such software must provide smooth access to, or better yet have integrated within it, device drivers for some or all of the following: image scanners, CD-ROM drives, laserdisc drives, and WORM drives for locally-obtained or locally-produced images. This software must also be capable of easily handling all major PC graphical file formats, including TIFF (Tag Image File Format), and EPS (Encapsulated PostScript), in addition to PCX.

Furthermore, it is suggested that such software be able to integrate one or more DBMSs (database management systems), beyond simply providing a DBMS gateway. In such a case, information could be approached by the user from the angle: "Give me a list of all ships with SS-N-22 launchers, ADMG-630 guns, AND amidships helipads", and when that list appears, the user could click on one of the names in the list to view the unit's image, and continue exploration from there using the "built-in" hypermedia links. Such a capability would facilitate implementation and integration of an interactive "unknown-unit-identifier" feature, similar to that outlined in the ideal GEOTREC of Chapter IV.

A further recommendation forthcoming from this research is that GEOTREC be designed in such a way that it could either be implemented as a standalone system, or integrated into computer systems which are currently in use in, or under development for, the Fleet.

Concerning the standalone implementation, the recommended host for such a system is a workstation. Although today's top-of-the-line PCs are extremely capable in terms of processing speed, primary memory capacity, and computational capabilities, they are still relatively limited in areas such as input/output (I/O) proficiency, particularly when compared to minicomputers and mainframes. "...in

terms of I/O capability and volume of external storage, microcomputers are far behind the mainframes, whereas for the processing capability and memory size the gap between the two types of computers is narrowing." ([Ref. 24], p.11) Because a "standalone mainframe" is obviously inappropriate for this application, a workstation host could be the ideal middle ground, providing a significant increase in I/O performance over PCs, especially in the retrieval and display of graphical images so central to an application like GEOTREC.

In addition to improved I/O effectiveness, workstations provide larger, higher-resolution screens than do standard PC configurations, permitting more detailed, higher-resolution images to be displayed and manipulated. These larger screens, coupled with architectures which are designed for greater primary memory utilization, allow workstations to provide more powerful windowing capabilities, perfect for hypermedia applications. Furthermore, increased primary memory will increase the effectiveness of gateways from the hypermedia software to other, related applications, without concern for the MS-DOS "640K" barrier.

While a standalone GEOTREC would be suitable in many situations, the application should also, ideally, be integratable with Fleet systems such as JOTS (Joint Operational Tactical System), FDDS (Flag Data Display System), OBU (OSIS [Ocean Surveillance Information System] Baseline Upgrade), and NIPS (Naval Intelligence Processing System) 2000, to name a few. In such a setting, GEOTREC could augment the existing system's capabilities by providing an on-line, hypermedia-based help and reference system. A typical scenario might take place in a FOSIC (Fleet Ocean Surveillance Information Center), where a watch officer, using OBU, receives a message on a new contact. To refresh his memory,

or to assist in preparing the next morning's flag briefing, the watchstander clicks on the OBU-displayed symbol which corresponds to the new contact and an image of that unit appears in a window on the OBU screen, with embedded hypermedia links which allow the user to select the type and amount of related information he wishes to explore.

A final recommendation of this thesis is that DOM-based threat recognition information of the type used in applications such as GEOTREC be published by appropriate naval and national intelligence organizations, such as DIA (Defense Intelligence Agency), NTIC (Naval Technical Intelligence Center), and the FICs (Fleet Intelligence Centers). For geographic information, DMA (the Defense Mapping Agency) already publishes a wide range of its products on DOM.

Paper publications should not necessarily be discontinued, as they do present an alternative for situations in which computer-based solutions are not practicable. However, distributing this information on DOM would provide an alternative which is superior to paper publications in a myriad of ways. One such benefit would be that a greater number of alternate views of a particular threat unit (not to mention full-motion video) could be supplied on DOM, over and above what would be practical to include in comparable paper publications. These additional views would significantly improve the user's perspective of how that unit will appear when seen in real life, leading to faster and more accurate threat recognition.

In order to facilitate the production of DOM by appropriate agencies, an incentive program could be instituted to encourage users to send in their locally-obtained or locally-developed digitized images, captured (e.g., scanned) on either

magnetic media or on WORM disks. This would provide a continuous source of "raw materials" in digitized form which could easily be edited and collated with other data into an updated DOM version. This updated DOM product could then be distributed (more easily and at far lower cost than paper) to all concerned, providing standardized, uniform dissemination throughout the Fleet.

In conclusion, threat recognition and geographical training are fundamental parts of a requisite knowledge base for a large number of naval personnel who are assigned to operational or operations-oriented support billets. And yet readiness in these areas is often lacking, in large part due to the paucity of readily available, motivational instruction tools. The development and deployment of a system such as GEOTREC would not only provide an enjoyable, intuitive, yet challenging way to foster multi-sensory learning, but also a quick, powerful, and easy-to-use reference to needed information for a myriad of operational scenarios.

LIST OF REFERENCES

1. Barrett, Edward, ed., *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information*, MIT Press, 1989.

2. Freedman, Alan, *The Computer Glossary*, The Computer Language Company, Inc., 1989.

3. Taylor, Nick G., "HyperShell Hypertext Control System (Product Description and Shareware Notice)", product documentation from Text Technology, Macclesfield, Cheshire, England, 1989.

4. Fersko-Weiss, Henry, "Hypertext Connects Disparate Data: Extract a World of Data, Layer by Layer", *Lotus*, vol.5, no.3, March 1989.

5. Bonner, Paul, and Leong, Kathy Chin, "Quantum Leap: The Smart Document", *PC Computing*, v.2, no.7, July 1989.

6. Naval Postgraduate School, Monterey, California, Report NPS52-88-010, *Managing Multimedia Data - An Exploration*, by Klaus Meyer-Wegener, Vincent Y. Lum, and C. Thomas Wu, March 1988.

7. Coale, Kristi, "The Body Electric", *MacUser*, vol.5, no.3, March 1989.

8. Harrison, Shelley, "The ABCs of Multi-Media CD-I & DVI", *CD-ROM EndUser*, vol.1, no.5, September 1989.

9. Jarrett, Lynn, "Versatility Helps HyperCard Make Its Mark in Media", *Digital Review*, vol.6, no.12, 27 March 1989.

10. Conklin, Jeff, "Hypertext: An Introduction and Survey", *IEEE Computer*, pp. 17-41, September 1987.

11. Naval Postgraduate School, Monterey, California, Report NPS52-87-050, *Integrating Advanced Techniques Into Multimedia DBMS*, by Vincent Y. Lum, C. Thomas Wu, and David K. Hsiao, November 1987.

12. Naval Postgraduate School, Monterey, California, Report NPS52-87-024, *Image Database Management In A Multimedia System*, by Klaus Meyer-Wegener, Vincent Y. Lum, and C. Thomas Wu, August 1988.

13. (no byline), "GECI International Challenges Apple's Hypercard with Hyperdoc for MS-DOS Machines", *Computergram International*, no.1138, 17 March 1989.

14. Rawles, Richard, "The Interface That Launched a Thousand Imitations", *MacWEEK*, vol.3, no.12, 21 March 1989.

15. Miller, David C., "Digital MultiMedia Era Opens: Through a Glass, DMMly", *CD-ROM EndUser*, vol.1, no.6, October 1989.

16. Taylor, Robert Ryland, *Conversion of Hard-Copy Documents to Digital Format Utilizing Optical Scanners and Optical Storage Media*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1989.

17. Leek, Matthew, "The MEDIA for CALS", *CD-ROM EndUser*, vol.1, no.8, December 1989.

18. Hutchison, Roger, "The ABCs of CD-ROM", *CD-ROM EndUser*, vol.1, no.5, September 1989.

19. (no byline), "Erasable? CD-ROM? or Both?", *CD-ROM EndUser*, vol.1, no.5, September 1989.

20. Alexander, Michael, "Everyone's Talking Multimedia", *Computerworld*, vol.23, no.36, 4 September 1989.

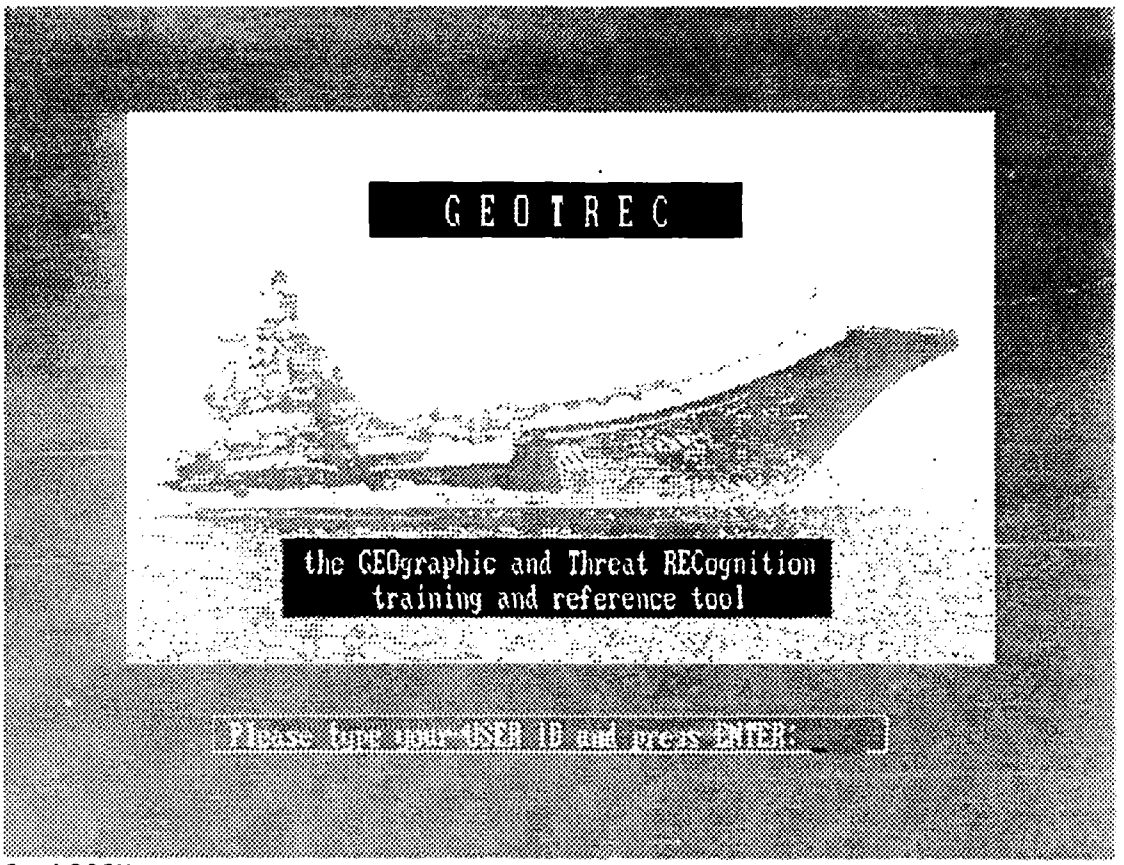
21. Karon, Paul, "KnowledgePro Links Two Technologies; Software Hybrid Merges Expert System and Hypermedia", *PC Week*, vol.6, no.15, 17 April 1989.

22. Morgan, Cynthia, "Toting Up Federal Micros: Analysis Finds 1.6 Million", *Government Computer News*, vol.9, no.1, 08 January 1989.

23. Polmar, Norman, *Guide To The Soviet Navy*, 4th ed., Naval Institute Press, 1988.

24. Naval Postgraduate School, Monterey, California, Department of Administrative Sciences, Working Paper No. 89-02, *Centralizing the Data, Distributing the Processing*, by Ran Giladi, and Moshe Zviran, January 1989.

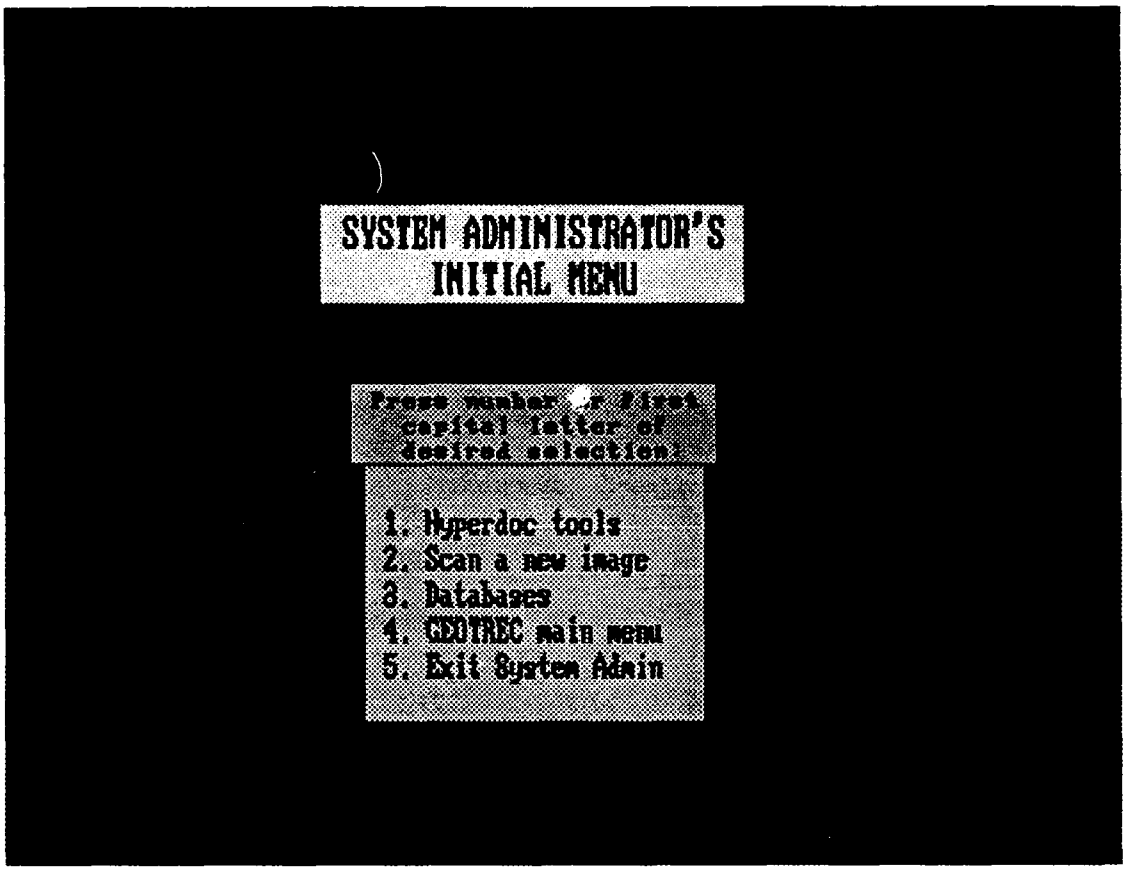
APPENDIX A: GEOTREC DEMONSTRATION



1 LOGIN screen

This appendix is intended to demonstrate GEOTREC's user interface, menu hierarchy and hypermedia capabilities. Obviously, it is difficult, if not impossible, to express in a linear report such as this the full feel and features of non-linear hypermedia. To assist in this, the screens are accompanied by a running textual description, written as if the reader and writer were both sitting at the computer terminal, exploring GEOTREC together.

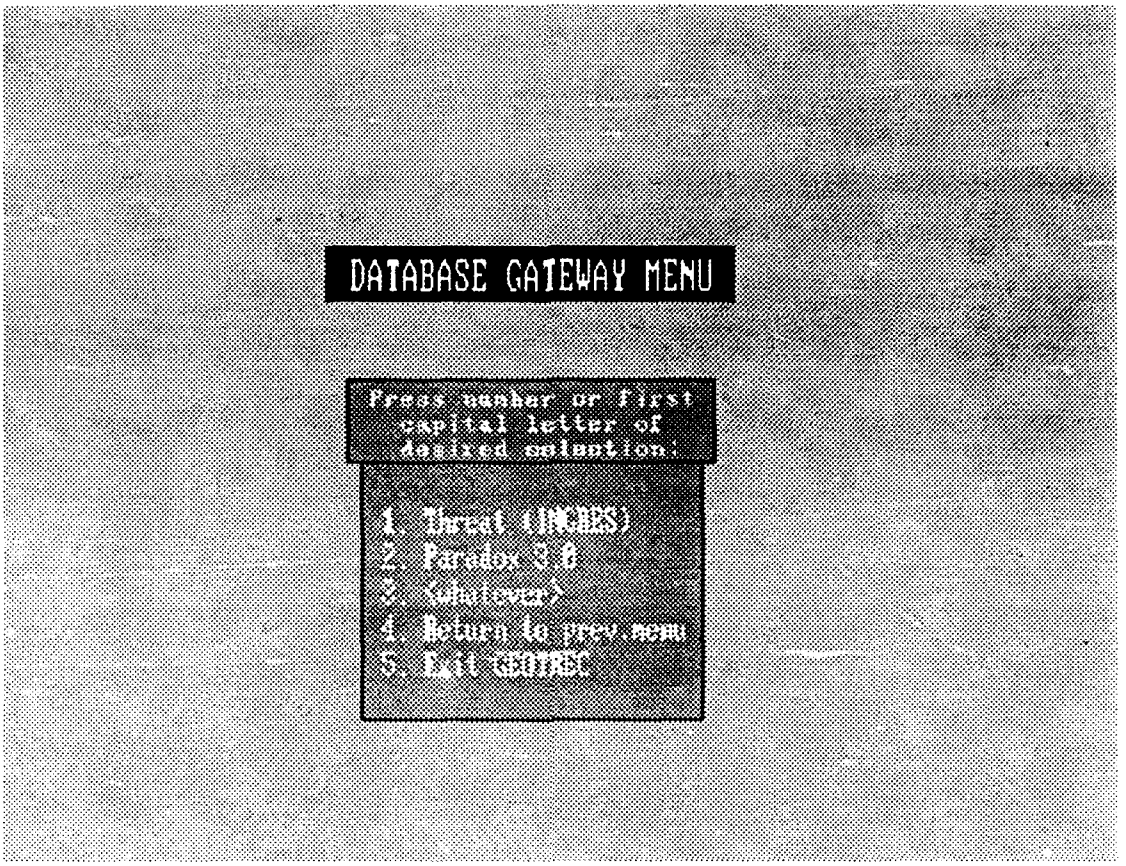
After the user starts GEOTREC, he is required to login with a user identification (USERID) as shown above. If the USERID is valid, the system prompts for a password to provide further security. In either case, the system will allow only three attempts for each; otherwise, the user is advised to seek assistance from the System Administrator, and is returned to DOS. If the USERID/password are the System Administrator's the System Administrator's Menu (screen 2) is accessed; for any other valid user, the Main Menu (screen 4) is displayed.



2 System Administrator's Menu

The System Administrator can access several functions not available to the ordinary user via the menu shown here, namely:

- "1. Hyperdoc Tools": provides access to all of the Hyperdoc object, graphic, text and other editors required to modify the Hyperdoc application;
- "2. Scan a new image": calls the scanner device driver software, from which the scanner can be operated to digitize new images and save them on the hard disk; from there, the images can be converted to Hyperdoc format, via Hyperdoc Tools, for use in the application and/or for writing to a WORM disk for more permanent storage (this scanner control feature was not implemented in this prototype);
- "3. Databases": calls the Database Gateway Menu (screen 3).



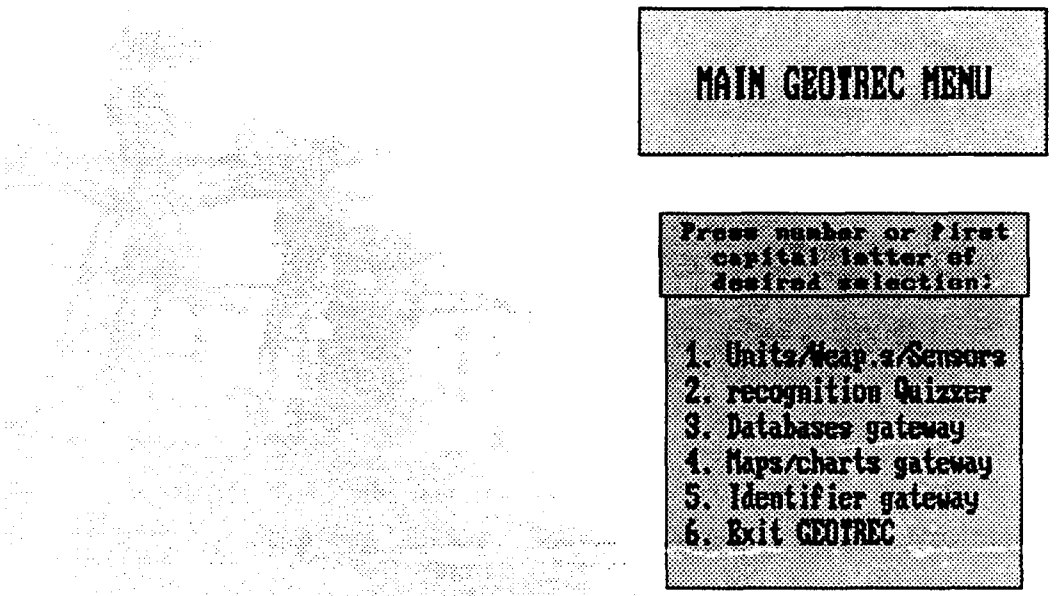
3 Database Gateway Menu, accessible from both the System Administrator's Initial Menu or the GEOTREC Main Menu.

The Database Gateways Menu, shown here, can be accessed from either the System Administrator's Initial Menu (screen 2), or the from the Main Menu (screen 4). Database packages such as Paradox 3.0 and an Ingres 5.0 application called the Threat Database[1] can be accessed from this menu.

Although not implemented in this application, the gateway mechanism would optimally provide the System Administrator access to the Database Management Systems (DBMS) packages with read and write privileges (e.g., by passing his USERID and password as parameters to the DBMS applications), while allowing read-only capabilities to all other users. This would prevent a separate, additional login to the DBMS application, as is presently required by the Threat Database (i.e., no mechanism exists for passing the USERID and password parameters from one application to another).

---

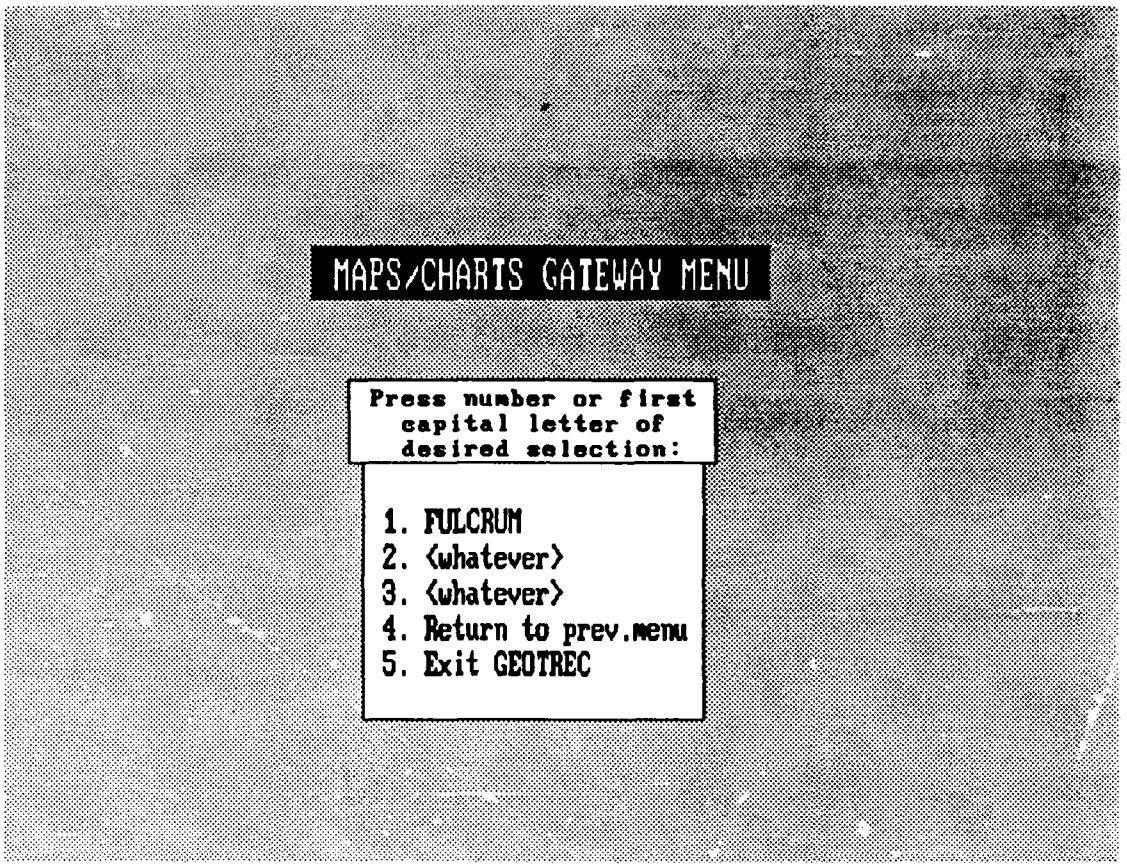
[1] The Threat Database was produced as a Naval Postgraduate School class project by Brad Triebwasser, Glenn Zeiders, and this author.



4 Main Menu

If the System Administrator selects "4. GEOTREC Main Menu" from his initial menu (screen 2), he will be at the same point at which the ordinary user would be after a valid login, namely, the Main Menu shown above. On this menu, the "6. Exit GEOTREC" option returns ordinary users to DOS (after asking for confirmation); for the System Administrator, it returns him to the System Administrator's Menu (screen 2). This gives the System Administrator flexibility to review modifications made to the application with Hyperdoc Tools, and then return to the Tools for further modifications without having to exit to DOS and re-login.

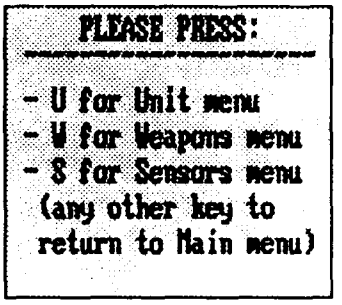
From the Main Menu, all users can access the major GEOTREC features shown. "3. Databases gateway" (screen 3) has been covered. "5. Identifier gateway" has not been implemented in this prototype application, but would be used to access a program which would perform the functions outlined in Chapter IV. The program would assist in the identification of unknown contacts via an interactive, query/response mechanism. The software would successively use the characteristics of the contact, provided by the user, to narrow the possibilities in its database, until a positive identification was made.



5 Maps and Charts Gateway Menu.

The Maps/Charts Gateway Menu, shown here, is displayed upon the selection of "5. Maps/Charts gateway" from GEOTREC's Main Menu (screen 4). From this menu, the user can access applications which allow the display and manipulation of various maps and charts. FULCRUM, the only such application presently available from the GEOTREC prototype, is a system which was developed at the Naval Ocean Systems Center (NOSC), San Diego, California.

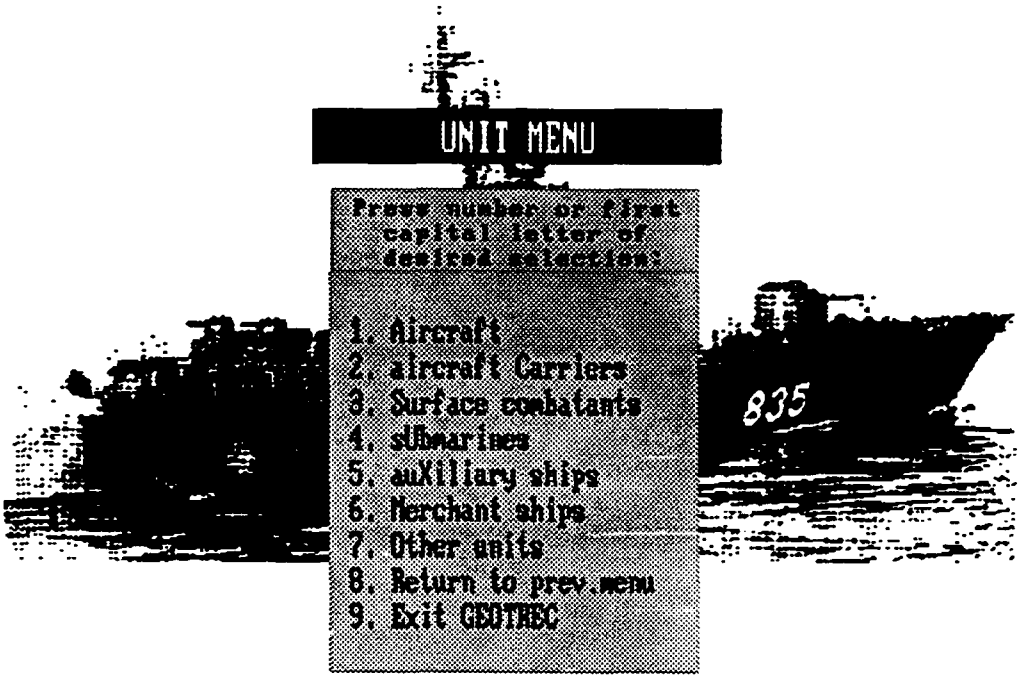
Among a wide range of other functions, FULCRUM's capabilities include the retrieval and display of digitized maps and charts which are stored on laserdiscs. FULCRUM allows the user to overlay annotations and data sets (e.g., unit positions, tracks, range circles/ellipses, and much more) on the geographic images, and would be invaluable as a geographic training and reference aid in the context of the GEOTREC application.



6 Main Menu after "1. Units/Weap.s/Sensors" option selected.

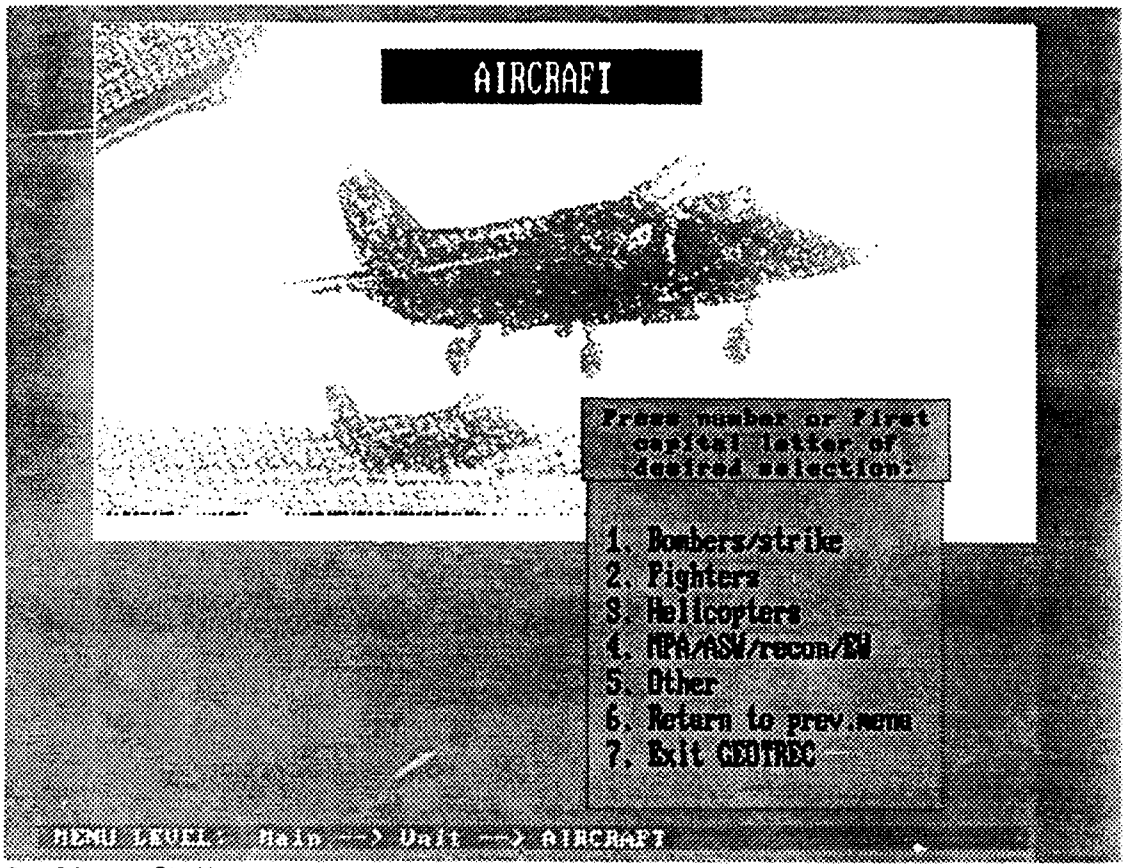
Let us now return to the Main Menu (screen 4), and select "1. Units/ Weap.s/Sensors". As shown above, the user is prompted to press the first letter of the desired selection: Units, Weapons or Sensors. Each of these three options provides access to the corresponding portion of GEOTREC's database of digitized images and information (a sort of on-line "Jane's All the World's..."). Although not implemented in this prototype, these images would optimally be stored on CD-ROM or laserdiscs, as opposed to on the computer's hard disk as presently done.

For further exploration, let's select "U" for Units and move on to screen 7.



7 Unit Menu

The Unit Menu, shown here, provides the user access to images of a wide variety of naval and navy-related units, ranging from aircraft to auxiliaries. Selecting "1. Aircraft" moves us on to screen 8.



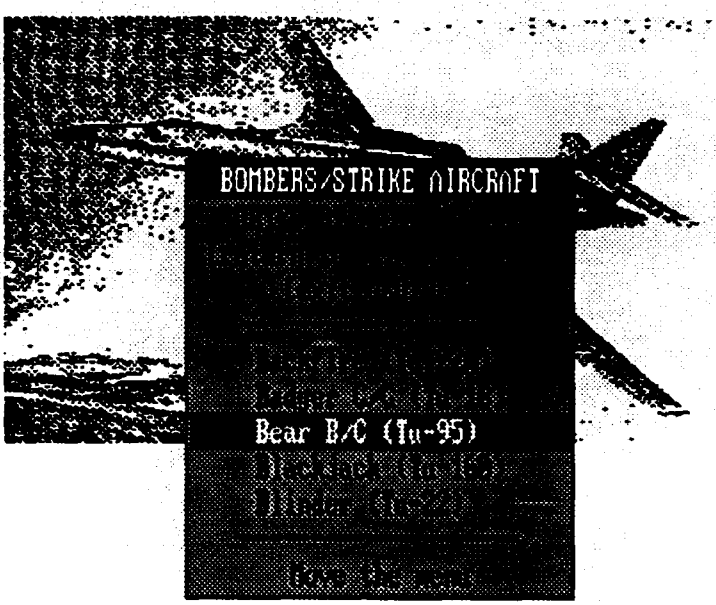
8 Aircraft Menu

In order to assist the user in making a selection, the Aircraft Menu, shown here, narrows the options further by breaking down the aircraft in the database by functional type. From here, let us select "1. Bombers/Strike Aircraft".

9 Bomber/Strike Aircraft Menu (initial)

We have now reached the last stage of the system of menus, and are ready to begin viewing images. As with all menus on the "bottom layer" of the menu hierarchy, the Bomber/Strike Aircraft Menu, shown here, has several differences from the "higher" menus.

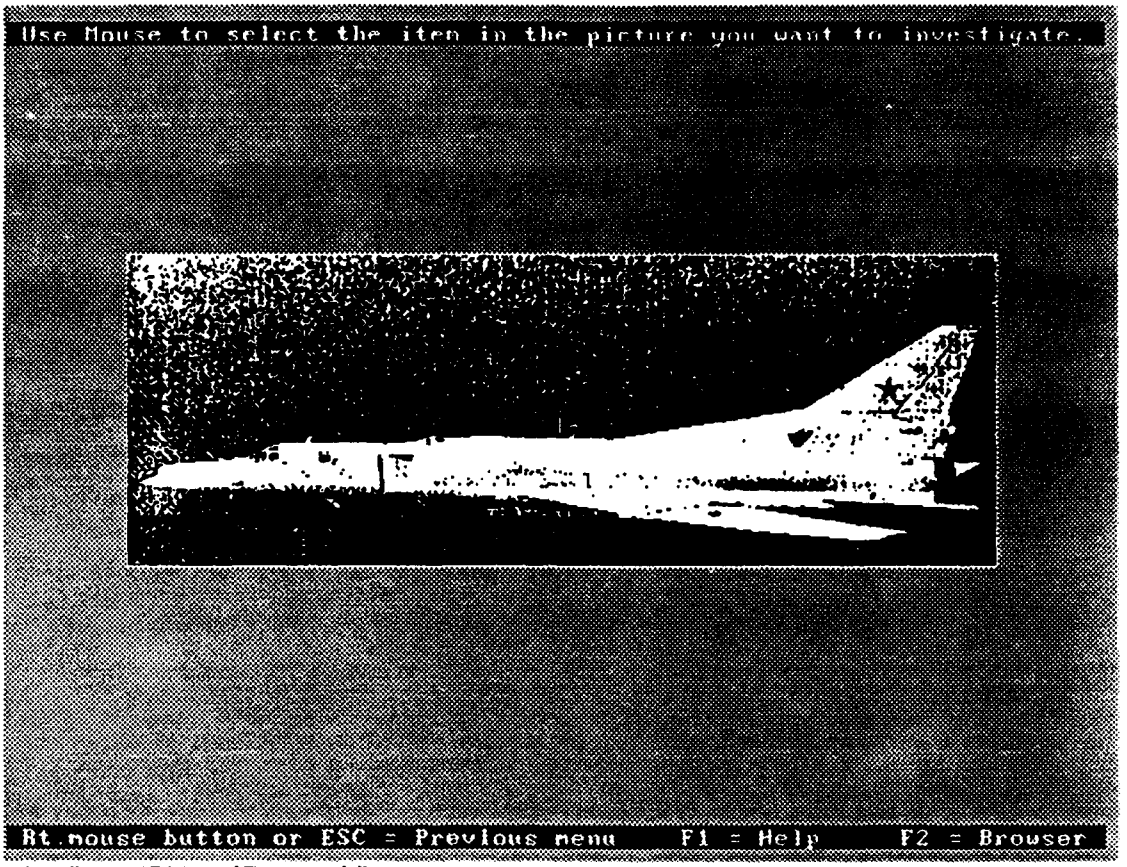
First, Hyperdoc's menuing subsystem is used, as opposed to code-created menus, in order to accommodate long and possibly changing lists of units. With this Hyperdoc menuing subsystem, the user can no longer press a number or the first capitalized letter of the desired selection; instead, the user must use the mouse or up/down arrow keys the highlight the proper choice, and then click the left mouse button or press the ENTER key. Although this can be somewhat less convenient to users, it does allow for the menu listing to be longer than the screen display, in which case the unseen selections will scroll into view as the arrow keys or mouse attempt to move to highlight them. In addition, a convenient "move the menu" option is automatically added at the end of all Hyperdoc menuing subsystem menus, allowing the user to view any portion of the screen that might be obstructed by the menu.



10 Bomber/Strike Aircraft Menu after "NATO Codename" has been selected for sorting method

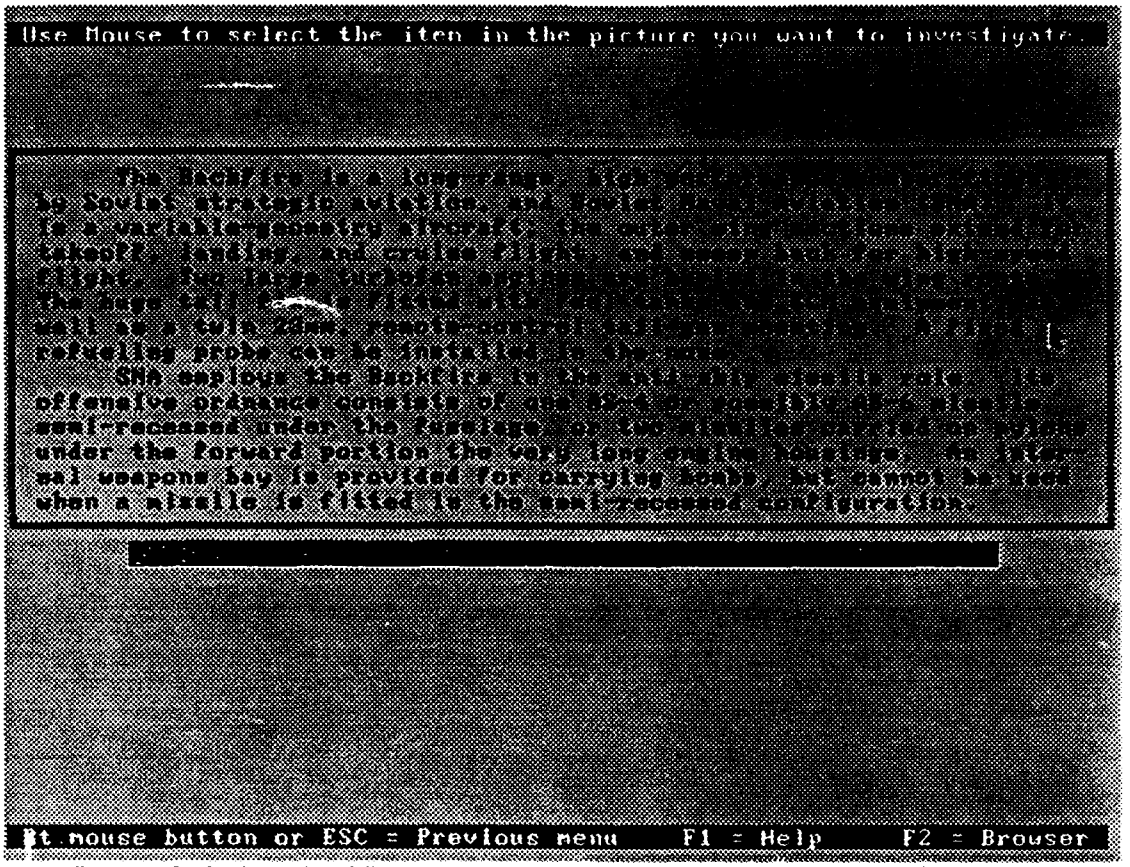
Second, the "bottom layer" menus provide the user with the option to view the list of applicable units, in this case Bomber/Strike Aircraft, sorted by either designator or by NATO Codename. IF we select "NATO Codename", the result is shown above (screen 10).

In order to demonstrate the hypermedia capabilities of the GEOTREC application, let's choose the first selection, "BACKFIRE (Tu-22m)".



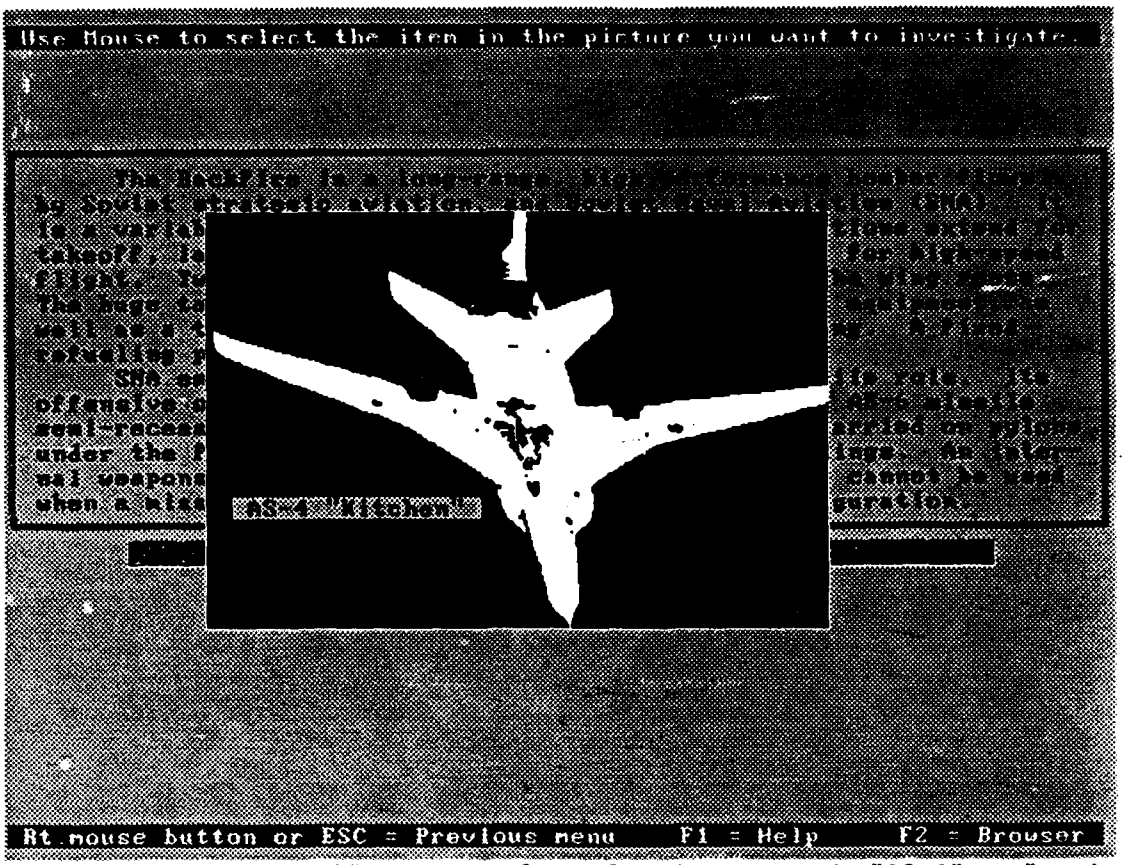
11 "BACKFIRE (Tu-22M)" screen

This image of a Tu-22m Backfire bomber is displayed in Hyperdoc's hypermedia mode invoked by the command "PROCESSDOC". In this mode, command nodes, known in Hyperdoc as objects, are associated ("linked") with certain locations ("hot spots", or "transparent buttons") on the image. If the user clicks with the mouse (as instructed at the top of the screen) on any of these buttons, the command that is linked to that location will be executed. In this case, the transparent button covers the entire fuselage of the aircraft (few other features are outstanding, due to the low image resolution forced by Hyperdoc's .PCX-only constraints). If we put the cursor on the fuselage and click the left mouse... (see next screen, screen 12)



12 "BACKFIRE (Tu-22M)" screen, after clicking on aircraft's fuselage to get more info

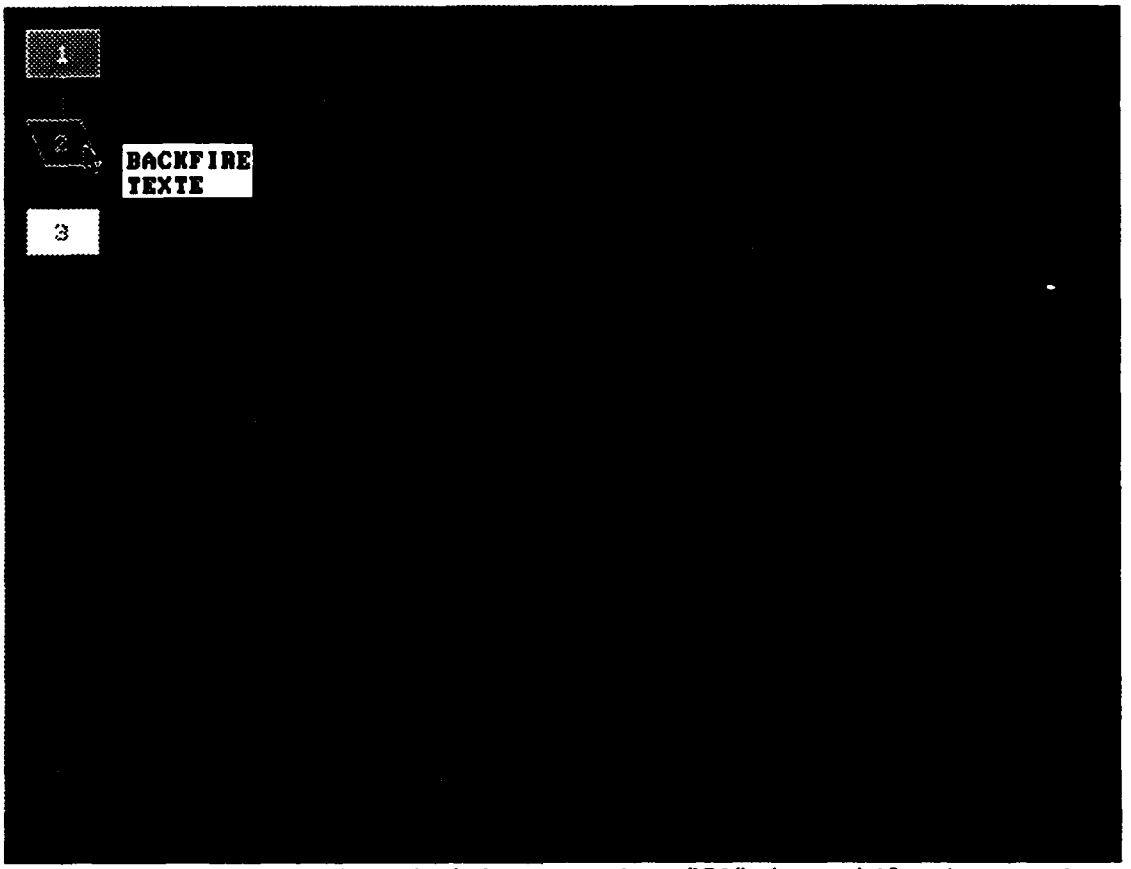
...a box of information about the Backfire is displayed. This text box is also covered with various hypermedia buttons. If the user, while reading, decides to seek more information on, say, the AS-4 missile mentioned in the text box, he simply has to point to that word with the cursor and click on it, and more information on the AS-4 is displayed --in this case,... (see next screen, screen 13)



13 "BACKFIRE (Tu-22M)" screen, after clicking on words "AS-4" or "semi-recessed under the fuselage"

...an image of the AS-4 carried by the Backfire in the semi-recessed underbelly position. For more information the user could just go on clicking.

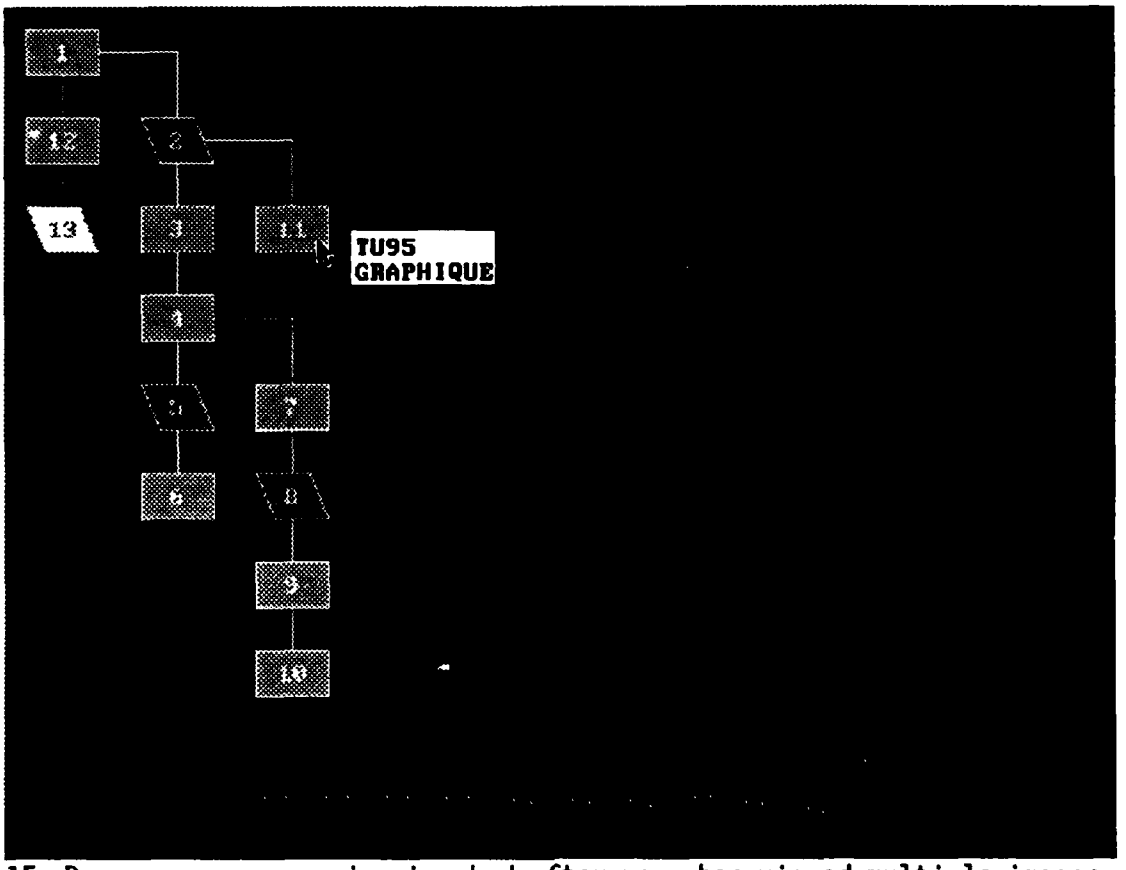
One could imagine, and rightly so, that with this continual clicking the user could quickly get "lost", unable to retrace his steps to get back to the original object at hand (which in this case was the Backfire bomber image). To assist in this, Hyperdoc has included a "browser", a kind of road-map feature initiated automatically with the first "PROCESSDOC" command. As indicated at the bottom of the screen, the browser is always available at the touch of a key -- the F2 key (see next screen).



14 Browser screen (invoked by pressing "F2" key while in previous screen)

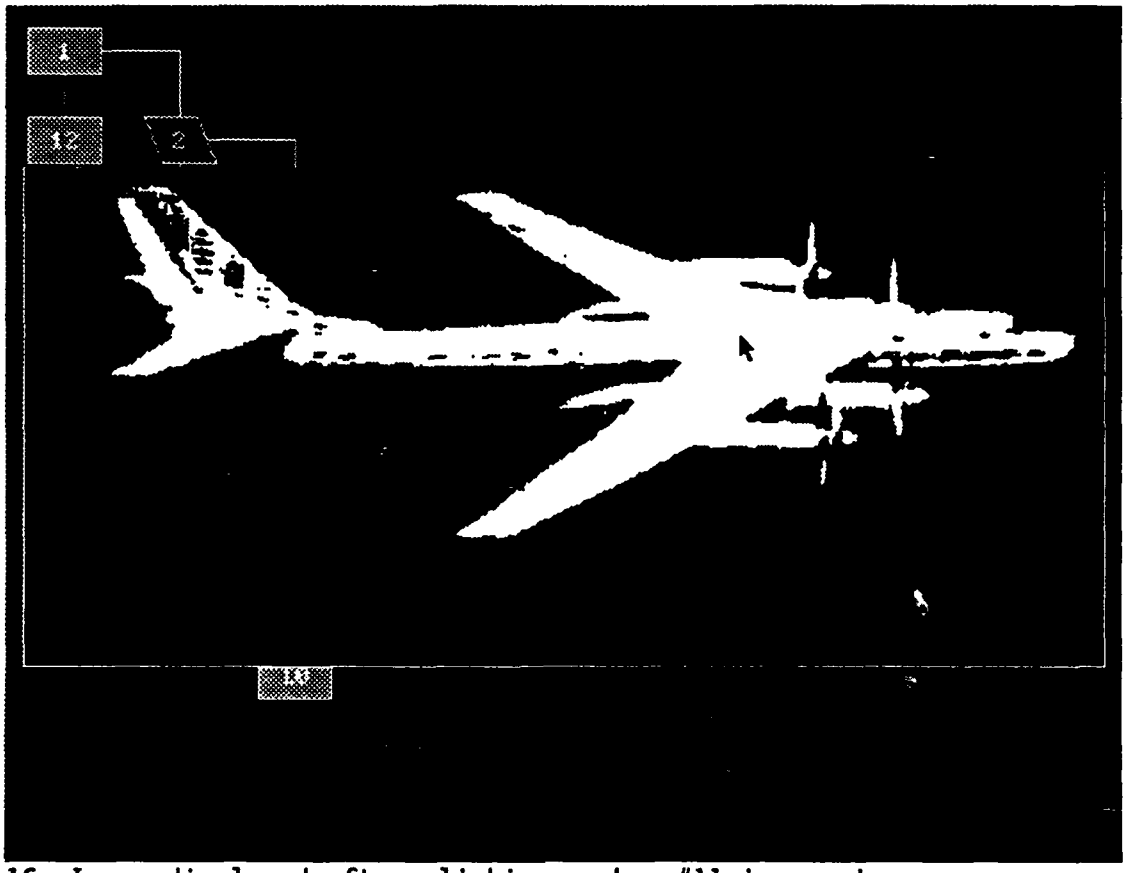
As seen here, the browser consists of a series of polygons linked by lines that represent the various screens or nodes that were linked upon successive user clicks; rectangles represent graphical images, parallelograms represent textual images.

Whenever a polygon is being pointed to by the cursor, an additional label box appears, indicating the name of that node (in this case, the failure of Hyperdoc to completely translate its software from the original French can be seen by the "e" on the word "Texte"; likewise, if a graphical rectangle is pointed to the word used by Hyperdoc is "Graphique" -- see next screen, screen 15)



15 Browser screen again, invoked after user has viewed multiple images.

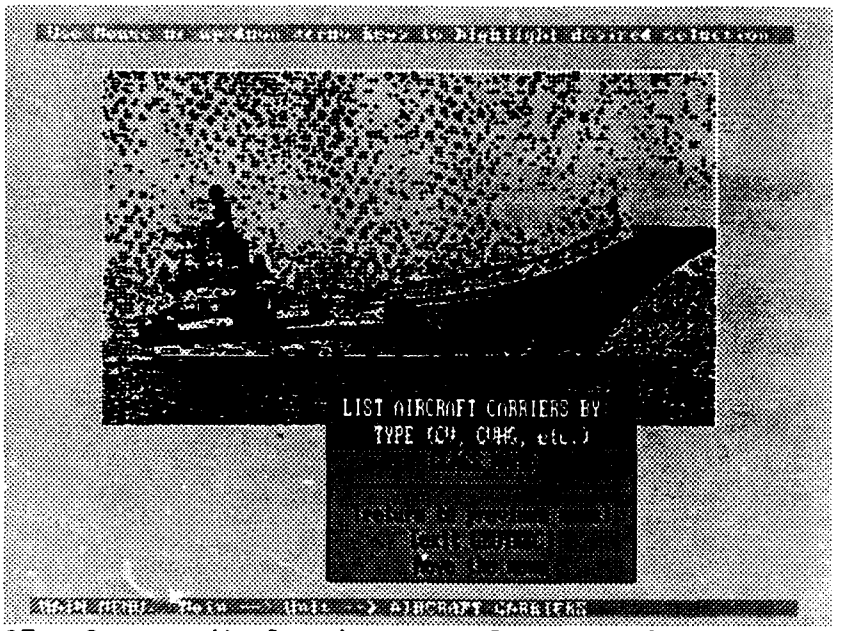
The browser does more than just map out the path taken by the user; it also allows the user to quickly return to any node in this "network" by simply clicking on that node. In this case, a click on the "TU-95 Graphique" node brings up... (see next screen, screen 16)



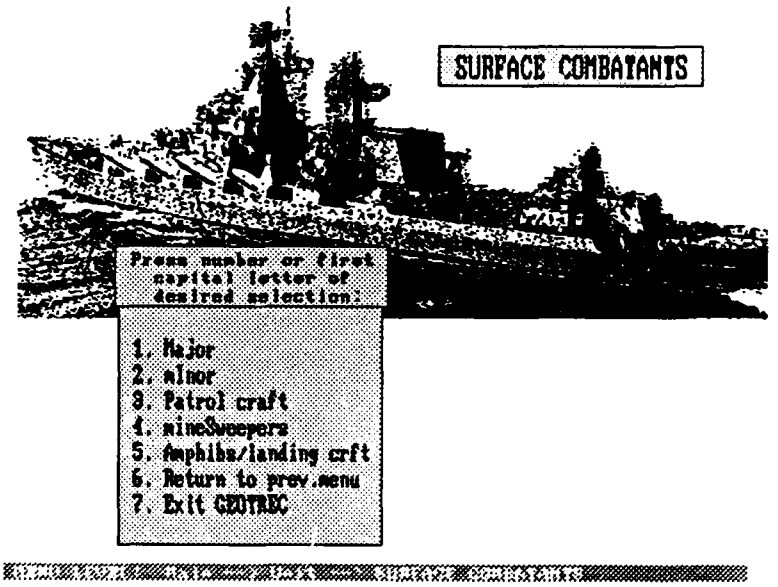
16 Image displayed after clicking on box #11 in previous screen

...the image of the Tu-95 Bear aircraft previously viewed. From this image, the user could go right on clicking, exploring additional features of the aircraft, and calling up the browser (with the F2 key) whenever needed. A press of the ESC key at any point during the browsing session will return the user to the Bomber/Strike Aircraft Menu (screen 9).

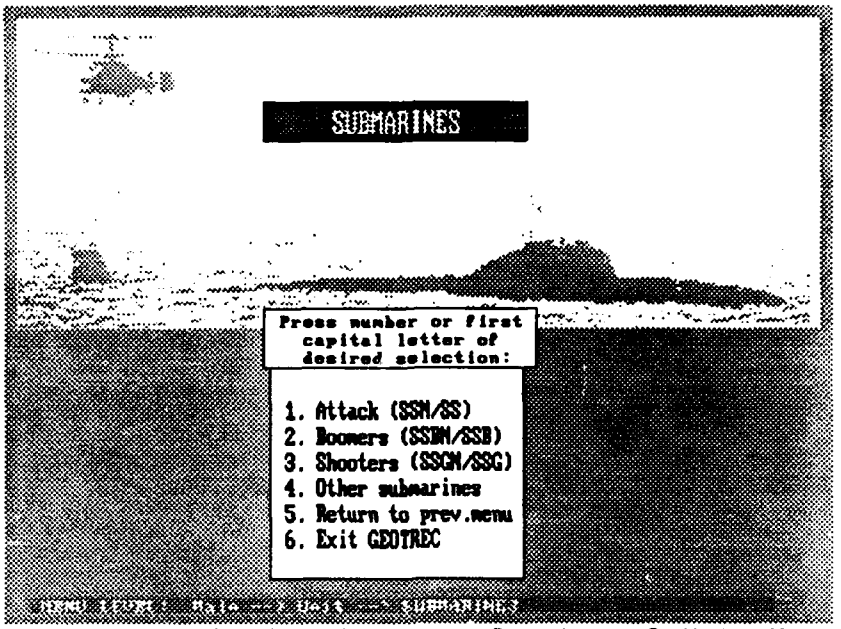
The same hypermedia features are available upon selecting any of the other options given in the Unit Menu (screen 7). The following pages show the menu screens presented for each of these options, so as to provide a feel for what else is available.



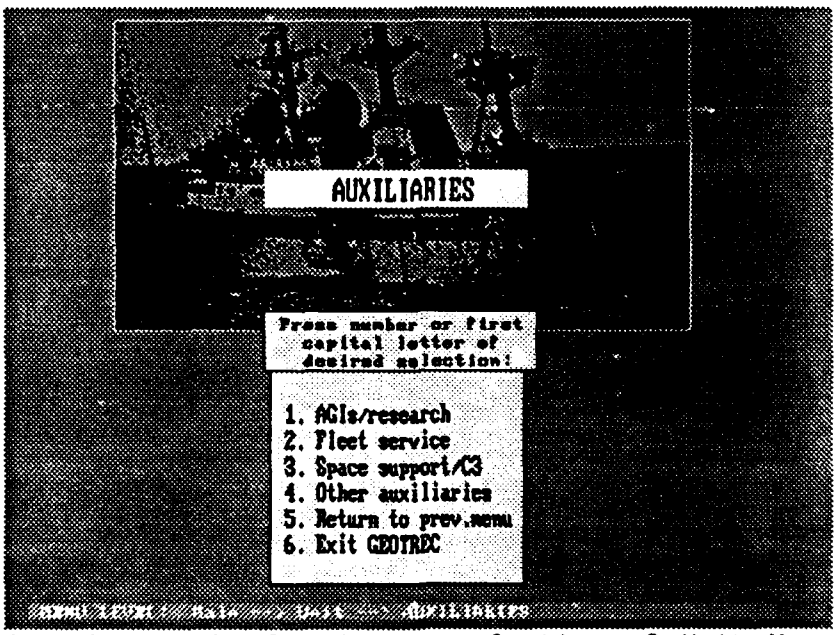
17 Screen displayed upon selection of Unit Menu choice "2. aircraft Carriers".



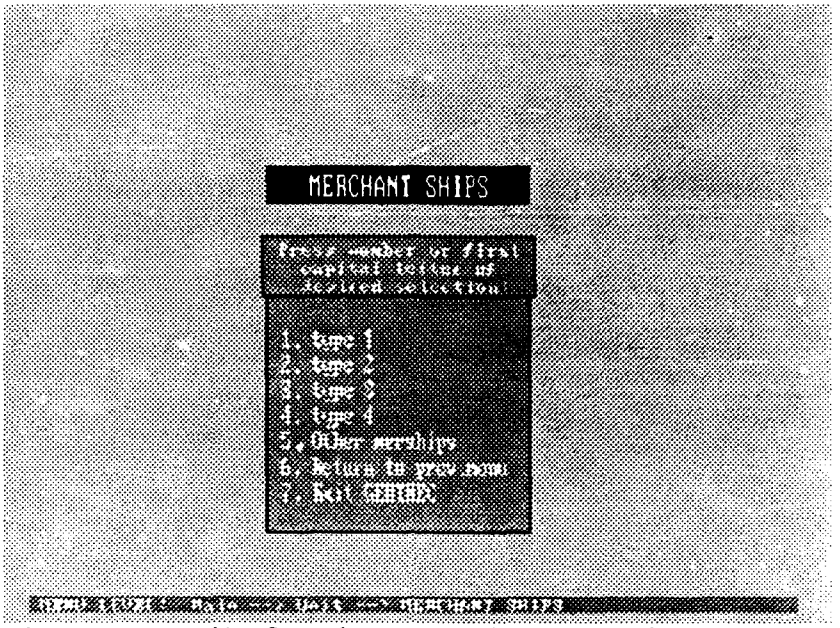
18 Screen displayed upon selection of Unit Menu choice "3. Surface combatants".



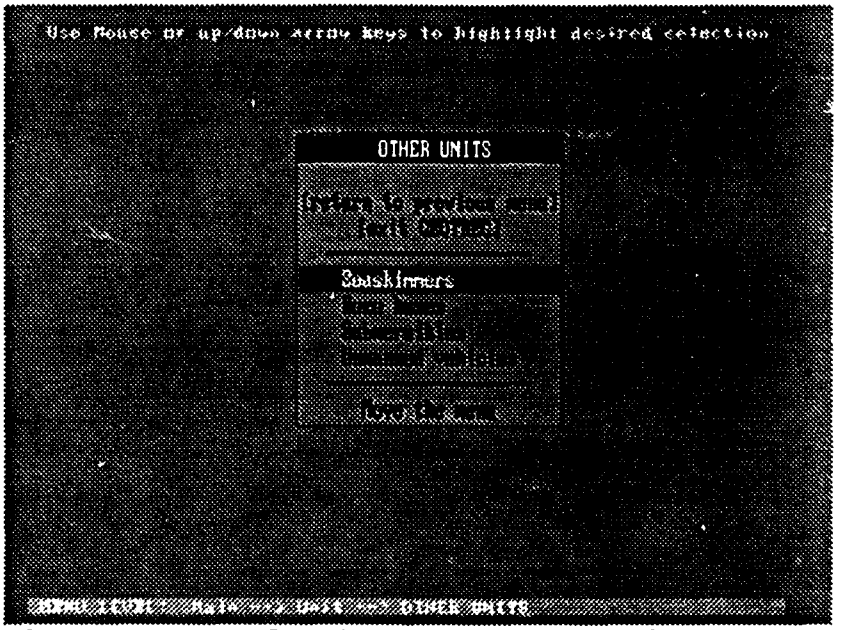
19 Screen displayed upon selection of Unit Menu choice "4. sUbmarines".



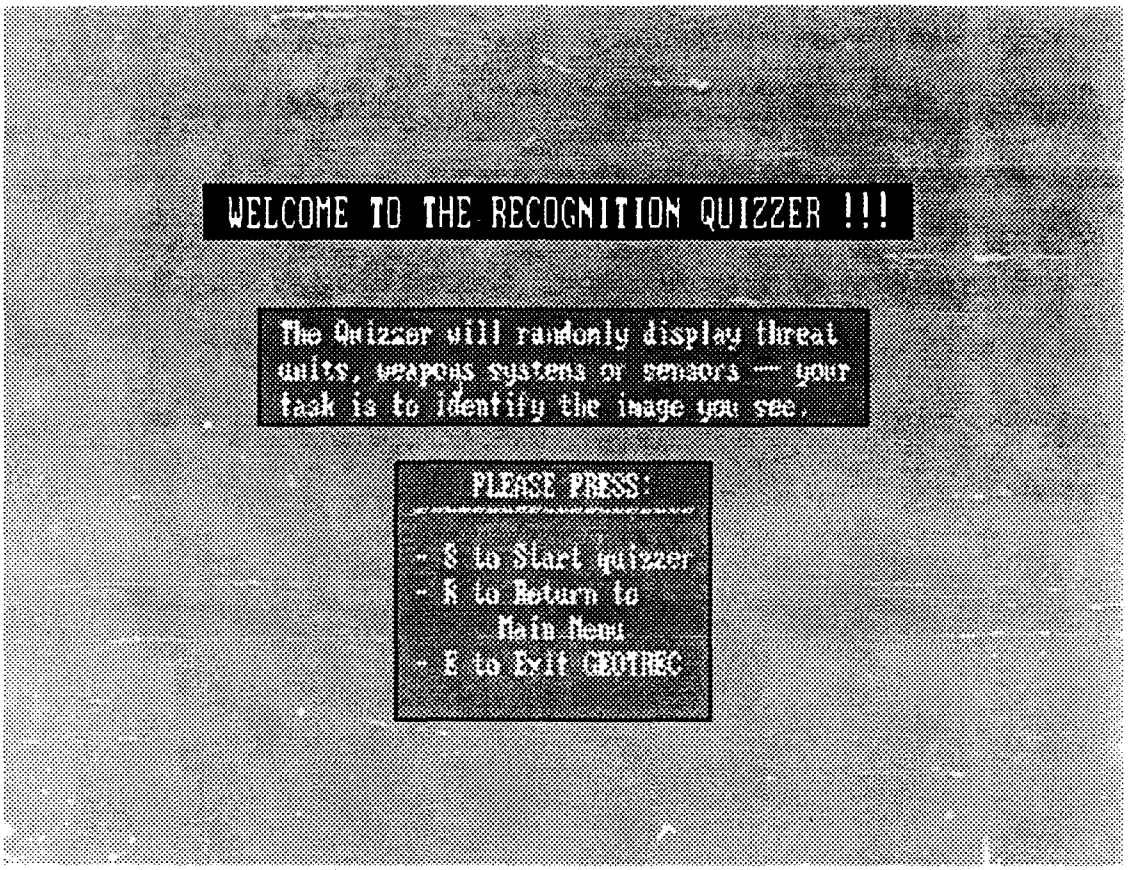
20 Screen displayed upon selection of Unit Menu choice "5. auXiliaries".



21 Screen displayed upon selection of Unit Menu choice "6. Merships".

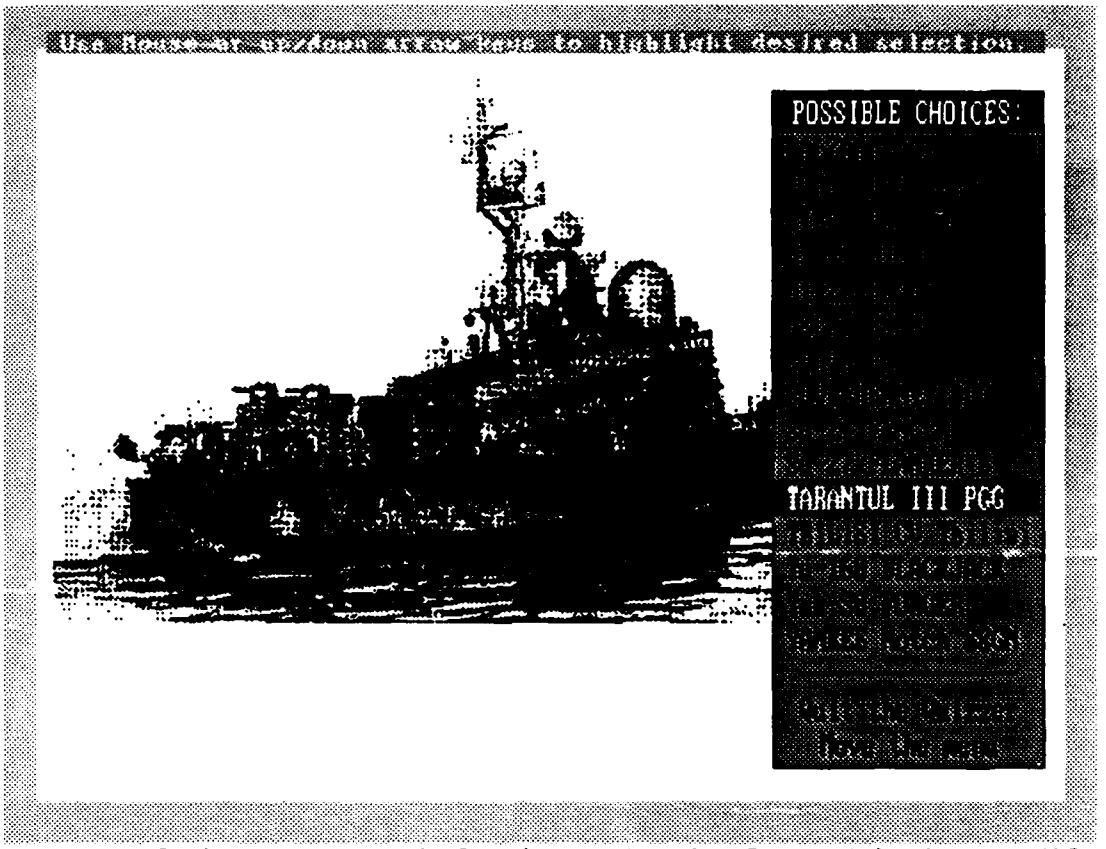


22 Screen displayed upon selection of Unit Menu choice "7. Other units".



23 Opening screen of QUIZZER (selected from Main Menu)

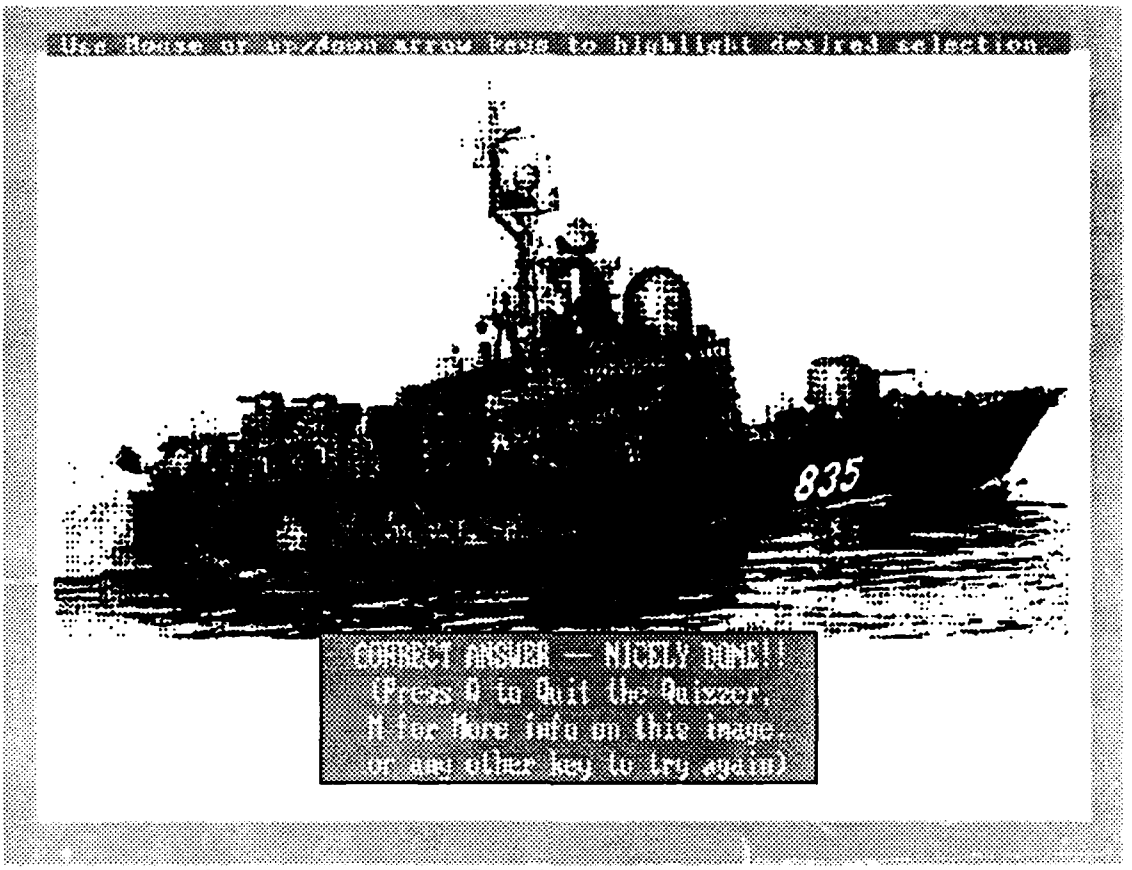
Let us now examine the final GEOTREC feature available from the Main Menu (screen 4), namely selection "2. recognition Quizzer". The Quizzer is a feature which, as described by the opening screen (shown above), will randomly display images of various threat units (could be modified to include just about anything, or, conversely, to display images of all one type, e.g., submarines only).



24 One of the images randomly chosen for display, with the possible choices listed on the menu at right.

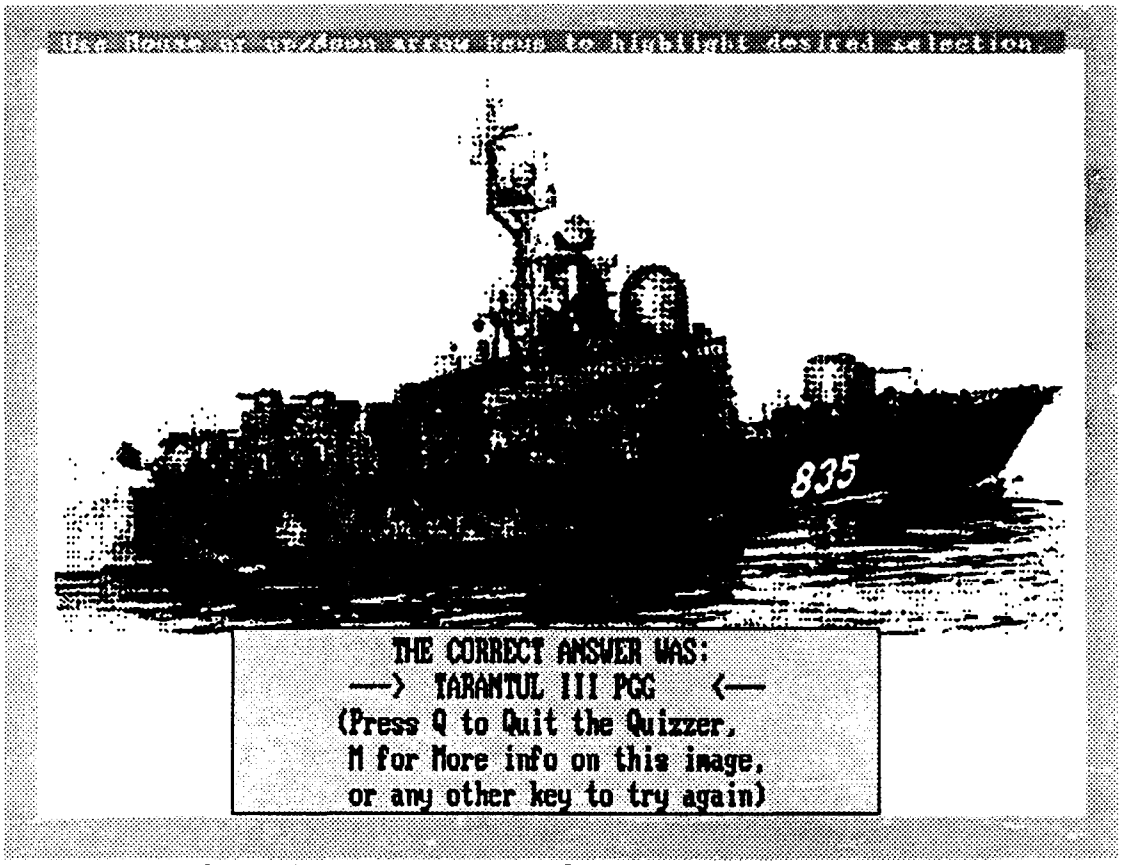
The first randomly displayed image is shown here, with the list of possible choices from which the user can select overlaid in a menu on the right. Again, the user can move and resize the menu as necessary, even to the point where only one menu selection at a time is visible -- and yet all selections would still be available, simply by scrolling up or down with the mouse or arrow keys.

The user is allowed three attempts to select the correct answer from the list of possible choices. If the user makes a correct selection on any of those three attempts, the following screen is displayed (see next screen, screen 25):



25 Screen after a correct selection made.

If a wrong selection is made, the Quizzer informs the user the selection was incorrect, and indicates how many attempts have been made. If the third attempt is also incorrect the following screen is displayed (see next screen, screen 26):

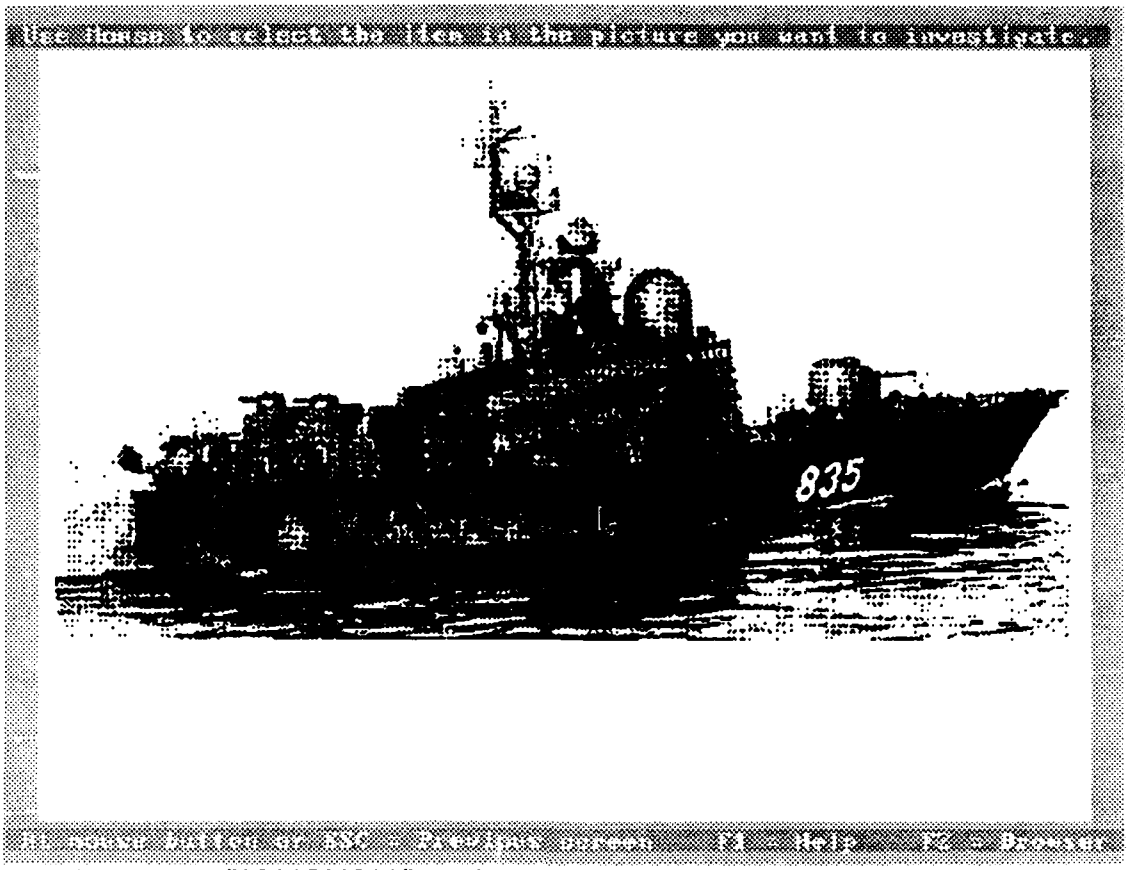


26 Screen after three incorrect selection attempts made.

Whether the user makes a correct selection or three successive incorrect selections, he is given three options:

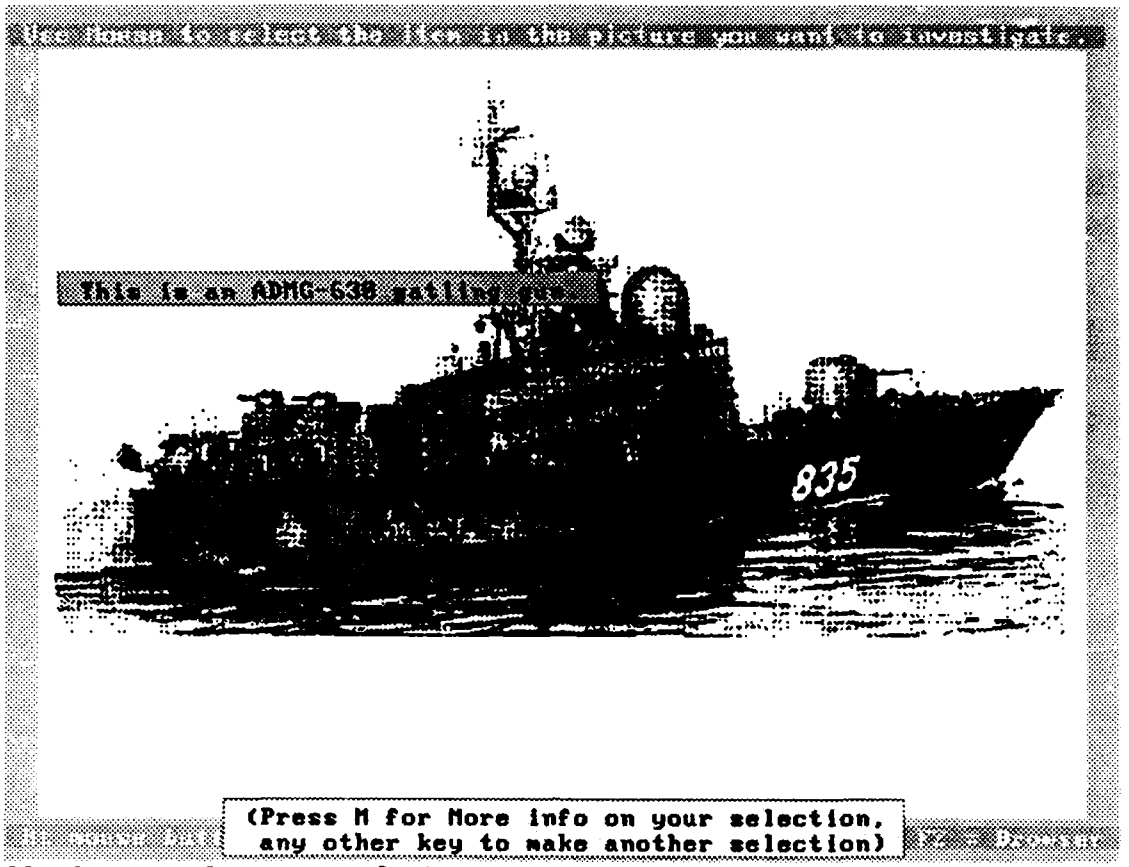
- press Q to quit the Quizzer and return to the Initial Quizzer Menu (screen 23);
- press M to get more information on the image being displayed; or,
- go on to another randomly selected image (by pressing any key).

Pressing M for More information will invoke Hyperdoc's hypermedia command "PROCESSDOC", as shown (see next screen, screen 27):



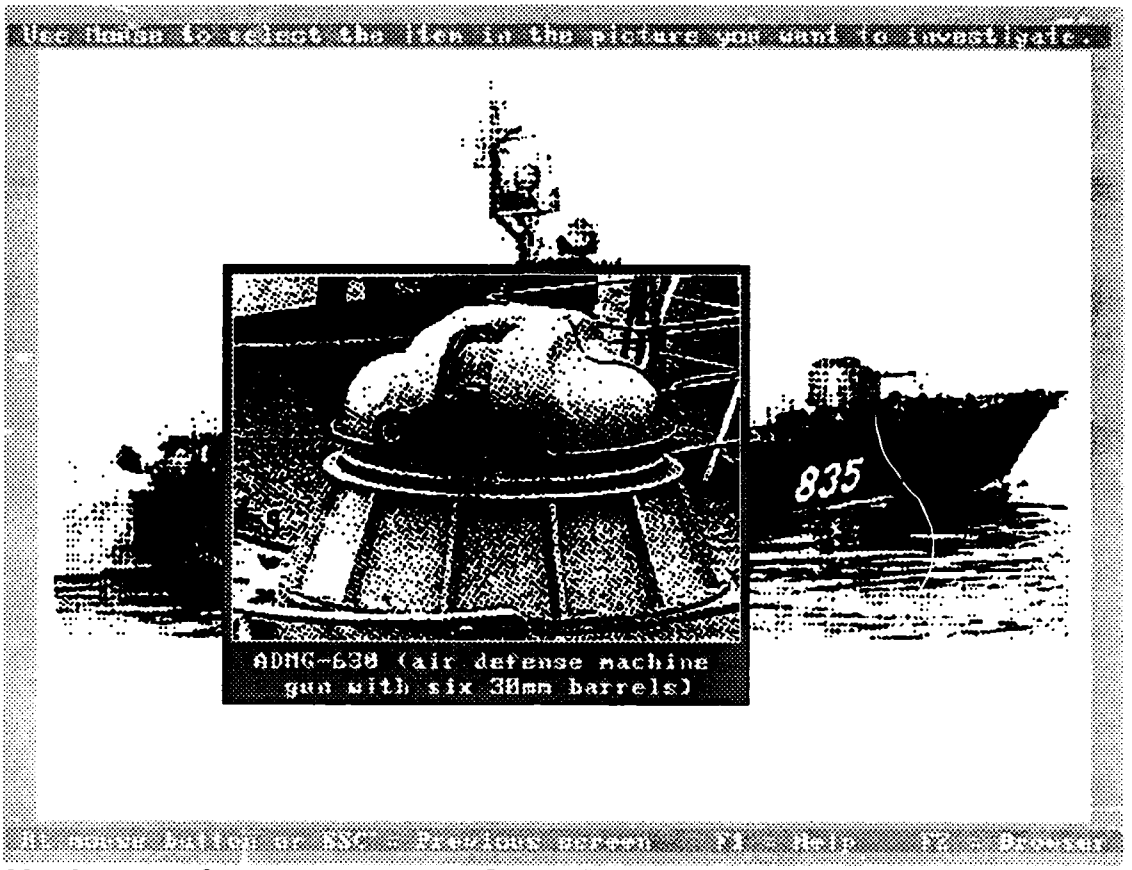
27 Image in "PROCESSDOC" mode, after user has pressed "M" (for More information) from either the correct or incorrect answer screens.

As seen before, the user simply needs to click on any major feature of the displayed image to receive more information about that feature. If, in this picture of a Tarantul III PGG (guided missile patrol boat), the user clicked on either of the two small guns near the aft end of the unit, he would see a label appear describing the gun as an ADMG-630 gatling gun, as shown in the next screen (screen 28):



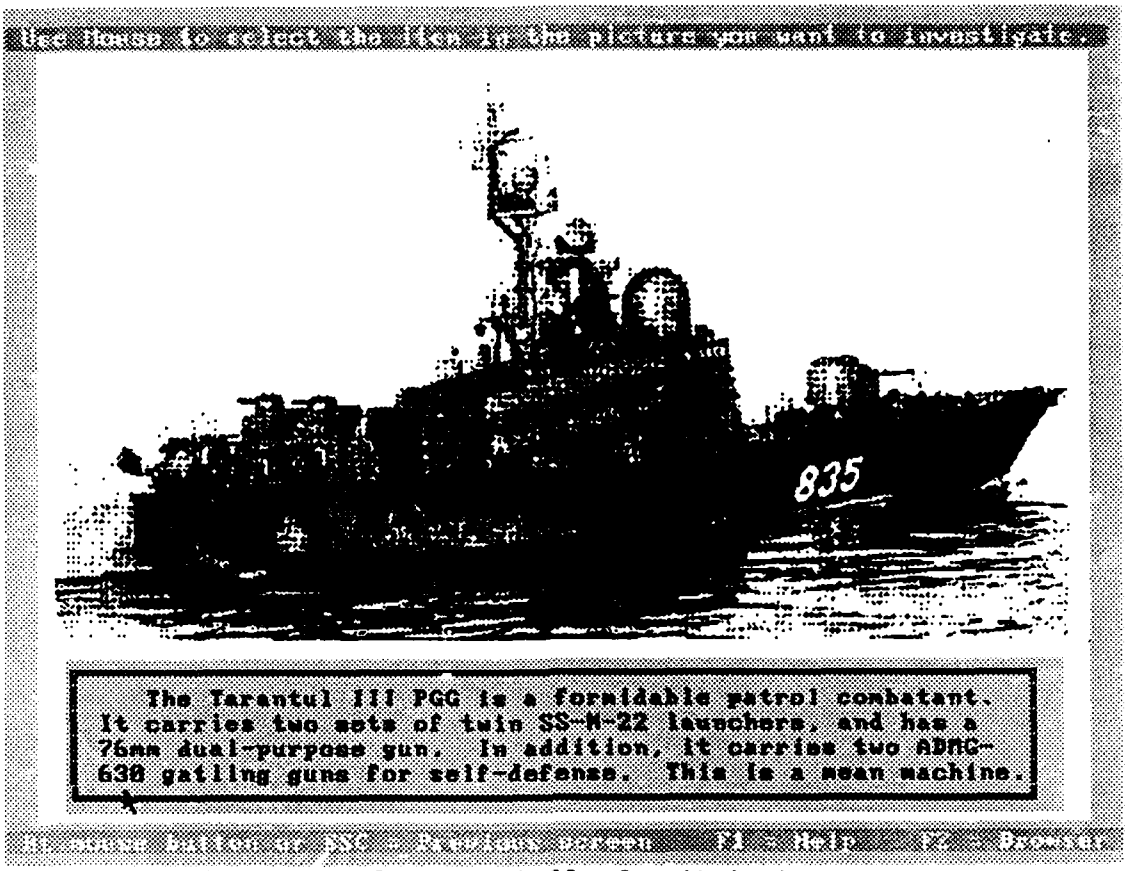
28 Screen after user clicks on a major feature (in this case, one of the two small guns on the unit's aft end).

In addition to the label, a yellow instruction box appears at the bottom of the screen, giving the user the option to press M for yet more information on his selection, or any other key to return to the image for further selections or exiting. If "M" is pressed for more information... (see next screen, screen 29)



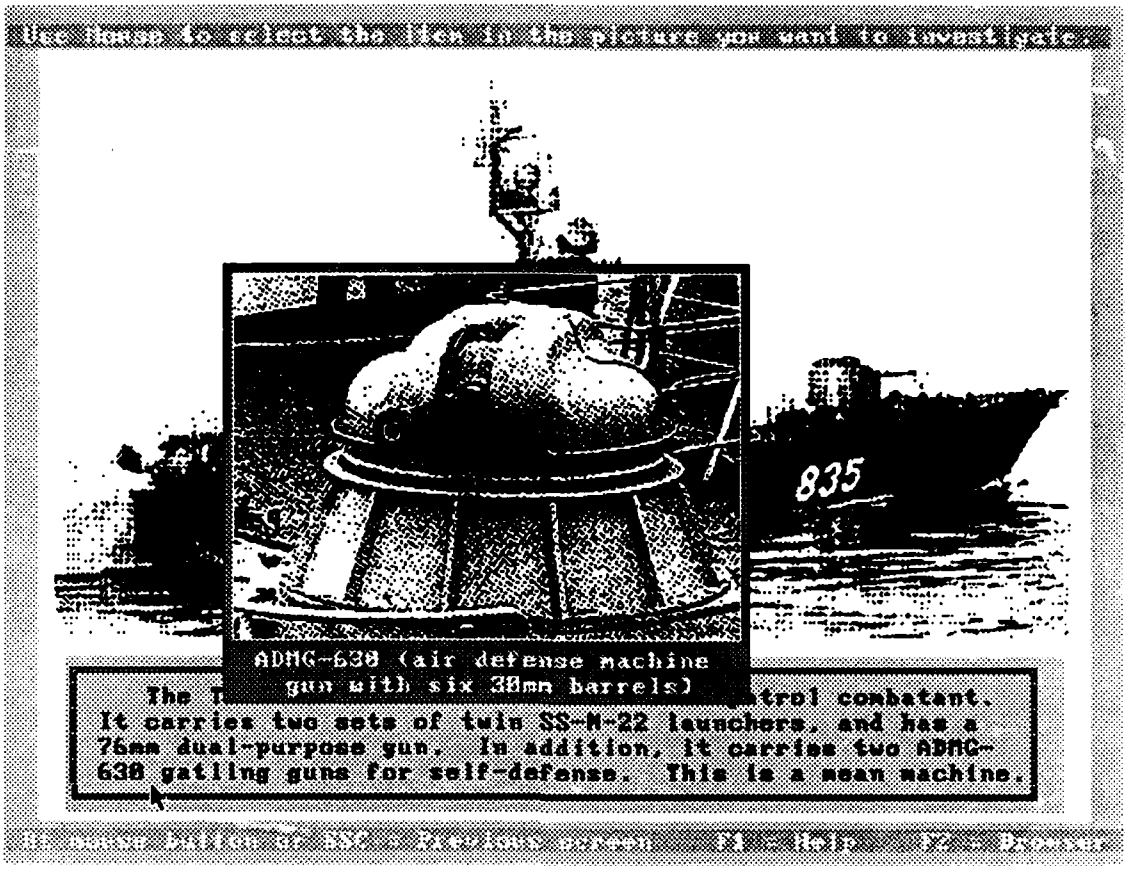
29 Screen after user again selects "M" for more information.

...a close-up image of the ADMG-630 air defense machine gun is displayed. And this is not the only route the user could take to get to this overlaid image; as shown on the next two pages (screens 30 and 31), the user could, from the initial Tarantul image (screen 24), click anywhere on the hull of the patrol boat to see...



30 Screen after user clicks on hull of unit in image.

...a text box, describing the patrol boat, as shown here. Then, to get more information about the ADMG-630 (or any of the other features mentioned in the text box), the user would simply click on that word to see displayed... (see next screen, screen 31)



31

...the same close-up view of that gun system. Once again, the clicking could continue if the user wanted to find even more information about the ADMG-630, or if even to explore other related information.

That brings us to the conclusion of this look at the features of this GEOTREC implementation. Keep in mind that GEOTREC was intended as a prototypical demonstration application, rather than an exhaustive study of all of possible capabilities. The hypermedia features demonstrated in this appendix show only a small portion of what can potentially be done, particularly as more robust and mature hypermedia software packages, including follow-on versions of Hyperdoc, become available.

APPENDIX B: HYPERDOC ACTION FILES (CODE)

A. LOGIN.YOB

```
/* 1: start LOGIN */
@logintrys = 0;
@sysadminvar = 0;
%CALL, "COMMON", 6;/*initializes vars for use in later menus*/
%CLEARSCREEN, 10/*lt.green*/;
%FONT, "System48.fnt";
%DISTEXT, &welcomewin, 135,85, 9/*black on lt.blue*/,
     "                          ",,
     "  WELCOME TO GEOTREC      ",,
     "                          ";
%DISDOC, "Tbilisi1", "G", 0, 4, 0,0, 0/*bkgnd on white*/,
     70,60, 570,370;
%FONT, "System96.fnt";
%DISTEXT, @geotrecwin, 210,100, 1/*white on blue*/,
     "    G E O T R E C    ";
%FONT, "System72.fnt";
%DISTEXT, @spellitoutwin, 160,300, 2/* white on green */,
     " the GEOgraphic and Threat RECognition ",,
     "      training and reference tool      ";
%EXECUTE, "Login", 2;                  /* call userid block */
END_OBJECT

/* 2: enter userid */
%FONT, "System72.fnt";
%QUESTTXTWIN, &userid,
  " Please type your USER ID and press ENTER: ", 120,400,
  4/*no.of spaces*/, 1/*force uprcase*/, 9/*blk on lt.blue*/;
%CHOICE, &userid;                    /* list of valid userids */
{
                                     /*"2468"=Sys.Administrator*/
     %CASE= "2468":  @logintrys = 0;
                     %EXECUTE, "Login", 3;
                                    /* all other qualified users*/
     %CASE= "ASDF":  @logintrys = 0;
                     %EXECUTE, "Login", 4;
     %CASE= "1234":  @logintrys = 0;
                     %EXECUTE, "Login", 4;
     %CASE= "A1B2":  @logintrys = 0;
                     %EXECUTE, "Login", 4;
     %CASE= "6789":  @logintrys = 0;
                     %EXECUTE, "Login", 4;
     %CASE= "WXYZ":  @logintrys = 0;
                     %EXECUTE, "Login", 4;
```

```
     %CASE= "WWWW":   @logintrys = 0;
                      %EXECUTE, "Login", 4;
                                /* if entry left blank... */
     %CASE= "":       %CALL, "Login", 6;
                                /* if invalid entry made...*/
     %DEFAULT:        @logintrys = @logintrys + 1;
                      %IF, @logintrys = 3;
                      {
                          %CALL, "Login", 7;
                      }/* end IF */
                      %ELSE;
                      {
                          %CALL, "Login", 8;
                      }/* end ELSE */
}/* end CHOICE */
%EXECUTE, "Login", 2;
END_OBJECT

/* 3: enter Password (for SYSTEM ADMINISTRATOR) */
%FONT, "System72.fnt";
%QUESTTXTWIN, &password,
 " Please type your PASSWORD and press ENTER: ", 110,400,
 8/*# of spaces*/, 1/*force uppercase*/, 9/*blk on lt.blue*/;
%CHOICE, &password;
{
                                /*"DALILAMA"= Sys.Admin'or*/
     %CASE= "DALILAMA":
                      @sysadminvar = 1;
                      %CLOSEWIN, @spellitoutwin;
                      %CLOSEWIN, @geotrecwin;
                      %EXECUTE, "SysAdmin", 1;
                                /*if entry left blank...*/
     %CASE= "":       %CALL, "Login", 6;
                                /*if invalid entry made...*/
     %DEFAULT:        @logintrys = @logintrys + 1;
                      %IF, @logintrys = 3;
                      {
                          %CALL, "Login", 7;
                      }/* end IF */
                      %ELSE;
                      {
                          %CALL, "Login", 9;
                      }/* end ELSE */
}/* end CHOICE */
%EXECUTE, "Login", 3;
END_OBJECT

/* 4: enter Password (for all other qualified users) */
%FONT, "System72.fnt";
%QUESTTXTWIN, &password,
 " Please type your PASSWORD and press ENTER: ", 110,400,
```

```
 8/*# of spaces*/, 1/*force uppercase*/, 14/*blk on yellow*/;
%CHOICE, &password;
{
                                        /*All other qualified users*/
     %CASE= "CAKEHOLE":   %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
     %CASE= "SPAMSPAM":   %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
     %CASE= "RATBAG":     %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
     %CASE= "AMIGADOG":   %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
     %CASE= "FOSTERS":    %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
     %CASE= "HOSEHEAD":   %CALL, "Login", 5;
                          %EXECUTE, "MainMenu", 1;
                                  /*if entry left blank... */
     %CASE= "":           %CALL, "Login", 6;
                                  /*if invalid entry made...*/
     %DEFAULT:            @logintrys = @logintrys + 1;
                          %IF, @logintrys = 3;
                          {
                              %CALL, "Login", 7;
                          }/* end IF */
                          %ELSE;
                          {
                              %CALL, "Login", 9;
                          }/* end ELSE */
}/* end CHOICE */
%EXECUTE, "Login", 4;
END_OBJECT

/* 5: Close windows */
%CLOSEWIN, @spellitoutwin;
%CLOSEWIN, @geotrecwin;
%RETURN;
END_OBJECT

/* 6: end GEOTREC */
%FONT, "System72.fnt",
%DISTEXTSTOP, &endgeotrec, 175,375, 8/*white on dark gray*/,
     "    ENTRY CANNOT BE LEFT BLANK !!  ",,
     "     Press E to Exit GEOTREC, or   ",,
     "     any other key to continue.    ";
%CHOICE, &endgeotrec;
{
     %CASE= 69/*'E" in ASCII*/:  %END;/*user ret'd to DOS*/
     %DEFAULT/*any other key*/:  %RETURN;
}/* end CHOICE */
END_OBJECT
```

```
/* 7: error msg -- too many login attempts */
%FONT, "System96.fnt";
%DISTEXT, @toomanywin, 130,350, 4,  /*red bkgnd*/
     " TOO MANY UNSUCCESSFUL LOGIN ATTEMPTS! ";
%WAIT,,2.5;
%FONT, "System72.fnt";
%DISTEXT, @sysadminwin, 130,385, 4, /*red bkgnd*/
     "  Please see the System Administrator for help.  ";
%WAIT,,2.5;
%FONT, "System96.fnt";
%WRITE, 220,440, 10/*lt.green ltrs*/,0/*black bkgnd*/,
     " NOW EXITING GEOTREC ";
%WAIT,,2.5;
%END;
END_OBJECT

/* 8: error msg -- invalid userid */
%FONT, "System72.fnt";
%DISTEXT, &invaliduseridwin, 180,375, 4/*white on red*/,
    "Invalid USER ID; please try again.";
%WAIT,,2.5;
%CLOSEWIN, &invaliduseridwin;
%RETURN;
END_OBJECT

/* 9: error msg -- invalid password */
%FONT, "System72.fnt";
%DISTEXT, &invaliduseridwin, 180,375, 4/*white on red*/,
    "Invalid PASSWORD; please try again.";
%WAIT,,2.5;
%CLOSEWIN, &invaliduseridwin;
%RETURN;
END_OBJECT
```

B. SYSADMIN.YOB

```
/* 1: start SysAdmin */
%CALL, "COMMON", 6; /* set all var.s = 1 for next menus */
%CLEARSCREEN, 1/*blue bkgnd*/;
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 180,110, 14/*black on yellow*/,
     " SYSTEM ADMINISTRATOR'S ",,
     "      INITIAL MENU      ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 197,210, 11/*black on lt.azure*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,256, 11/* black on lt.azure */,
   "                          ",,
   " 1. Hyperdoc tools",,    /* 1 = ASCII 49; H = ASCII 72 */
   " 2. Scan a new image",,  /* 2 = ASCII 50; S = ASCII 83 */
   " 3. Databases       ",,  /* 3 = ASCII 51; D = ASCII 68 */
   " 4. GEOTREC main menu",, /* 4 = ASCII 52; G = ASCII 71 */
   " 5. Exit System Admin",, /* 5 = ASCII 53; E = ASCII 69 */
   "                          ";
%CALL, "COMMON", 5;                    /* closes all windows */
@callerobj = "SysAdmin";/*so DBMSMENU'll know calling obj*/
%CHOICE, &choicevar;
{
                                       /* "1. Hyperdoc tools" */
     %CASE= 49:                        /* "1" in ASCII */
               %EXECUTE, "SysAdmin", 2;
     %CASE= 72:                        /* "H" in ASCII */
               %EXECUTE, "SysAdmin", 2;

                                       /* "2. Scan a new image"*/
     %CASE= 50:                        /* "2" in ASCII */
               %EXECUTE, "SysAdmin", 3;
     %CASE= 83:                        /* "S" in ASCII */
               %EXECUTE, "SysAdmin", 3;

                                       /* "3. Databases" */
     %CASE= 51:                        /* "3" in ASCII */
               %EXECUTE, "DBMSMenu", 1;
     %CASE= 68:                        /* "D" in ASCII */
               %EXECUTE, "DBMSMenu", 1;

                                       /*"4. GEOTREC main menu"*/
     %CASE= 52:                        /* "4" in ASCII */
               %EXECUTE, "MainMenu", 1;
     %CASE= 71:                        /* "G" in ASCII */
               %EXECUTE, "MainMenu", 1;
```

```
                                        /* "5. Exit GEOTREC" */
     %CASE= 53:                         /* "5" in ASCII */
                %CALL, "COMMON", 4;
     %CASE= 69:                         /* "E" in ASCII */
                %CALL, "COMMON", 4;

     %DEFAULT:                          /* Invalid selection */
               %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "SysAdmin", 1;
END_OBJECT

/* 2: Choice 1 "Hyperdoc tools" */
%DOSEXECUTE, "c:\\command.com", "/c", "c:\\hdtools.bat";
END_OBJECT

/* 3: Choice 2 "Scan a new image" */
%DISTEXT, &scanwin, 125,250, 7/*black on lt.grey*/,
     " a SCANNER DRIVER will be called at this point ";
%WAIT,,2.0;
%CLOSEWIN, &scanwin;
%EXECUTE, "SysAdmin", 1;
END_OBJECT
```

C. DBMSMENU.YOB

```
/* 1: start DBMSMenu */
%CALL, "COMMON", 6; /* set all variables = 1 for next menus*/
%CLEARSCREEN, 12/*lt.red bkgnd*/;
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 185,135, 1/*white on blue*/,
     " DATABASE GATEWAY MENU ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 197,210, 9/*white on lt.blue*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,256, 9/*white on lt.blue */,
   "                          ",,
   " 1. Threat (INGRES)",,     /* 1 = ASCII 49; T = ASCII 84*/
   " 2. Paradox 3.0",,         /* 2 = ASCII 50; P = ASCII 80*/
   " 3. <whatever>",,           /* 3 = ASCII 51; */
   " 4. Return to prev.menu",,/* 4 = ASCII 52; R = ASCII 82*/
   " 5. Exit GEOTREC",,        /* 5 = ASCII 53; E = ASCII 69*/
   "                         ";
%CALL, "COMMON", 5;                    /* closes all windows */
%CHOICE, &choicevar;
{
                                       /* "1. Threat (INGRES)"*/
     %CASE= 49:                        /* "1" in ASCII */
               %EXECUTE, "DBMSMenu", 2;
     %CASE= 84:                        /* "T" in ASCII */
               %EXECUTE, "DBMSMenu", 2;

                                       /* "2. Paradox 3.0" */
     %CASE= 50:                        /* "2" in ASCII */
               %EXECUTE, "DBMSMenu", 3;
     %CASE= 80:                        /* "P" in ASCII */
               %EXECUTE, "DBMSMenu", 3;

                                       /* "3. <whatever>"*/
     %CASE= 51:                        /* "3" in ASCII */
               %CALL, "GENERIC", 3;
               %EXECUTE, "DBMSMenu", 1;

                                    /*"4.Returntoprev.menu"*/
     %CASE= 52:                        /* "4" in ASCII */
               %EXECUTE, @callerobj, 1;
     %CASE= 82:                        /* "R" in ASCII */
               %EXECUTE, @callerobj, 1;

                                       /* "5. Exit GEOTREC" */
     %CASE= 53:                        /* "5" in ASCII */
               %CALL, "COMMON", 4;
```

```
     %CASE= 69:                          /* "E" in ASCII */
                %CALL, "COMMON", 4;

     %DEFAULT:                           /* Invalid selection */
                %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "DBMSMenu", 1;
END_OBJECT

/* 2: Choice 1 "Threat (INGRES)" */
%DOSEXECUTE, "c:\\command.com", "/c", "c:\\ingres.bat";
END_OBJECT

/* 3: Choice 2 "Paradox 3.0" */
%DOSEXECUTE, "c:\\command.com", "/c", "c:\\paradox.bat";
END_OBJECT
```

D. MAINMENU.YOB

```
/* 1: MainMenu */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 15/*white bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%IF, @disdoc = 1;
{
    %DISDOC, "Udal_Mag", "G";
    @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 365,45, 13/*black on lt.magenta*/,
    "                          ",,
    "   MAIN GEOTREC MENU      ",,
    "                          ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 377,160, 11/*black on lt.azure*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 380,206, 11/*black on lt.azure*/,
"                          ",,
" 1. Units/Weap.s/Sensors",,   /* 1 = ASCII 49; U = ASCII 85*/
" 2. recognition Quizzer",,    /* 2 = ASCII 50; Q = ASCII 81*/
" 3. Databases gateway",,      /* 3 = ASCII 51; D = ASCII 68*/
" 4. Maps/charts gateway",,    /* 4 = ASCII 52; M = ASCII 77*/
" 5. Identifier gateway",,     /* 5 = ASCII 53; I = ASCII 73*/
" 6. Exit GEOTREC",,           /* 6 = ASCII 54; E = ASCII 69*/
"                        ";
%CALL, "COMMON", 5;                    /*close all windows*/
@callerobj = "MainMenu";/*so DBMSMenu'll know calling obj. */
%CHOICE, &choicevar;
{
                                       /* "1. Us/Ws/Ss" */
     %CASE= 49:                        /* "1" in ASCII */
               %CALL, "MainMenu", 3;
               %CALL, "COMMON", 6;/* set vars to = 1 */
               %EXECUTE, @uwsmenu, 1;
     %CASE= 85:                        /* "U" in ASCII */
               %CALL, "MainMenu", 3;
               %CALL, "COMMON", 6;/* set vars to = 1 */
               %EXECUTE, @uwsmenu, 1;

                                       /* "2. Recce Quizzer" */
     %CASE= 50:                        /* "2" in ASCII */
               @quizcalledbymain = 1;
               %EXECUTE, "Quizzer", 1;
```

```
      %CASE= 81:                          /* "Q" in ASCII */
                  @quizcalledbymain = 1;
                  %EXECUTE, "Quizzer", 1;

                                          /*"3. Database gateway"*/
      %CASE= 51:                          /* "3" in ASCII */
                  %CALL, "COMMON", 6;/* set vars to = 1 */
                  %EXECUTE, "DbmsMenu", 1;
      %CASE= 68:                          /* "D" in ASCII */
                  %CALL, "COMMON", 6;/* set vars to = 1 */
                  %EXECUTE, "DbmsMenu", 1;

                                          /* "4. Maps gateway" */
      %CASE= 52:                          /* "4" in ASCII */
                  %CALL, "COMMON", 6;/* set vars to = 1 */
                  %EXECUTE, "MapsMenu", 1;
      %CASE= 77:                          /* "M" in ASCII */
                  %CALL, "COMMON", 6;/* set vars to = 1 */
                  %EXECUTE, "MapsMenu", 1;

                                        /*"5. Identifier gateway"*/
      %CASE= 53:                          /* "5" in ASCII */
                  %CALL, "MainMenu" 4;
      %CASE= 73:                          /* "I" in ASCII */
                  %CALL, "MainMenu",4;

                                          /* "6. Exit GEOTREC" */
      %CASE= 54:                          /* "6" in ASCII */
                  %EXECUTE,"MainMenu",2/*spec.exit confirm*/;

      %CASE= 69:                          /* "E" in ASCII */
                  %EXECUTE,"MainMenu",2/*spec.exit confirm*/;

      %DEFAULT:                           /* Invalid selection */
                  %CALL, "COMMON", 3;
                  @clearscreen = 0;
}/* end CHOICE */
%EXECUTE, "MainMenu", 1;
END_OBJECT

/* 2: special exit confirm -- return to SYSADMIN */
%FONT, "System72.fnt";
%QUESTTXTWIN, &reply,
  " ARE YOU SURE YOU WANT TO EXIT? (y/n):  ", 150,250,
  1/*no.chars in reply*/, 1/*force uprcase*/, 4/*wt on red*/;
%IF, &reply = "Y";          /*if user REALLY wants to exit...*/
{
     %IF, @sysadminvar = 1;  /*if user is Sys.Administrator,*/
     {
         %EXECUTE, "SysAdmin", 1; /*return to SysAdmin menu.*/
     }/*end if*/
```

```
     %ELSE;                        /* if user is NOT Sys.Admin,*/
     {
         %END;                              /* exit from GEOTREC.*/
     }/*end else*/
}/*end if*/
%ELSE;                          /*if user DOESN'T wanna exit...*/
{
     @clearscreen = 0;
     %EXECUTE, "MainMenu", 1;          /* return to MainMenu*/
}/*end else*/
END_OBJECT

/* 3: Units/Weap.s/Sensors choice */
%DISTEXTSTOP, &uwschoice, 385,160, 13/*black on lt.magenta*/,
     "     PLEASE PRESS:      ",,
     " ~~~~~~~~~~~~~~~~~~~~ ",,
     " - U for Unit menu",,    /* U = ASCII 85 */
     " - W for Weapons menu",,/* W = ASCII 87 */
     " - S for Sensors menu",,/* S = ASCII 83 */
     "  (any other key to",,
     "  return to Main menu) ",,
     "                       ";
%CHOICE, &uwschoice;
{
     %CASE= 85:  @uwsmenu = "UnitMenu";
     %CASE= 87:  @uwsmenu = "WeapMenu";
     %CASE= 83:  @clearscreen = 1;
                 %CALL, "GENERIC", 3;
                 %EXECUTE, "MainMenu", 1;
     %DEFAULT:   %EXECUTE, "MainMenu", 1;
}/* end CHOICE */
%RETURN;
END_OBJECT

/* 4: in lieu of Identifier gateway */
%DISTEXTSTOP, &idgatewaywin, 175,175, 13/*blk on lt.magenta*/,
     " ...at this point, a gateway would be called,  ",,
     " starting an application which would assist in ",,
     " identifying unknown units via an interactive  ",,
     " Query/Response session with the user...       ",,
     "           PRESS ANY KEY TO CONTINUE           ";
%CHOICE, &idgatewaywin;
{
     DEFAULT:    %RETURN;
}/* end CHOICE */
END_OBJECT
```

E. MAPSMENU.YOB

```
/* 1: start MapsMenu */
%CALL, "COMMON", 6; /* set all variables = 1 for next menus*/
%CLEARSCREEN, 11/*chestnut bkgnd*/;
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 175,135, 0/*white on black*/,
     " MAPS/CHARTS GATEWAY MENU ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 197,210, 15/*black on white*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,256, 15/*black on white */,
   "                          ",,
   " 1. FULCRUM",,             /* 1 = ASCII 49; F = ASCII 70*/
   " 2. <whatever>",,          /* 2 = ASCII 50; */
   " 3. <whatever>",,          /* 3 = ASCII 51; */
   " 4. Return to prev.menu",,/* 4 = ASCII 52; R = ASCII 82*/
   " 5. Exit GEOTREC",,        /* 5 = ASCII 53; E = ASCII 69*/
   "                         ";
%CALL, "COMMON", 5;                  /* closes all windows*/
%CHOICE, &choicevar;
{
                                     /* "1. FULCRUM" */
     %CASE= 49:                      /* "1" in ASCII */
               %EXECUTE, "MapsMenu", 2;
     %CASE= 70:                      /* "F" in ASCII */
               %EXECUTE, "MapsMenu", 2;

                                     /* "2. Paradox 3.0" */
     %CASE= 50:                      /* "2" in ASCII */
               %CALL, "GENERIC", 3;
               %EXECUTE, "MapsMenu", 1;

                                     /* "3. <whatever>"*/
     %CASE= 51:                      /* "3" in ASCII */
               %CALL, "GENERIC", 3;
               %EXECUTE, "MapsMenu", 1;

                                   /*"4.Return to prev.menu"*/
     %CASE= 52:                      /* "4" in ASCII */
               %EXECUTE, "MainMenu", 1;
     %CASE= 82:                      /* "R" in ASCII */
               %EXECUTE, "MainMenu", 1;

                                     /* "5. Exit GEOTREC" */
     %CASE= 53:                      /* "5" in ASCII */
               %CALL, "COMMON", 4;
     %CASE= 69:                      /* "E" in ASCII */
```

```
                %CALL, "COMMON", 4;

     %DEFAULT:                          /* Invalid selection */
                %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "MapsMenu", 1;
END_OBJECT

/* 2: Choice 1 "FULCRUM" */
%DOSEXECUTE, "c:\\command.com", "/c", "c:\\fulcrum.bat";
END_OBJECT
```

F. UNITMENU.YOB

```
/* 1: Unit Menu */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 15/*white bkgnd*/;
     @clearscreen = 0;
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> UNIT",
     "                                          ";
%IF, @disdoc = 1;              /* only if called from above */
{
    %DISDOC, "Tarantul", "G", 1, 1, -10,0, 15/*blk on bkgnd*/,
         0,25, 640,450;
    @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 196,100, 1/*black on blue*/,
     "       UNIT MENU       ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 206,145, 10/*black on lt.green*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,191, 10/*black on lt.green*/,
   "                          ",,
   " 1. Aircraft",,          /* 1 = ASCII 49; A = ASCII 65 */
   " 2. aircraft Carriers",,/* 2 = ASCII 50; C = ASCII 67 */
   " 3. Surface combatants",,/*3 = ASCII 51; S = ASCII 83 */
   " 4. sUbmarines",,        /* 4 = ASCII 52; U = ASCII 85 */
   " 5. auXiliary ships",,   /* 5 = ASCII 53; X = ASCII 88 */
   " 6. Merchant ships",,    /* 6 = ASCII 54; M = ASCII 77 */
   " 7. Other units",,       /* 7 = ASCII 55; O = ASCII 79 */
   " 8. Return to prev.menu",,/*8= ASCII 56; R = ASCII 82 */
   " 9. Exit GEOTREC",,      /* 9 = ASCII 57; E = ASCII 69 */
   "                           ";
%CALL, "COMMON", 5;                    /* close all windows */
@callerobj = "UnitMenu";/* so GENERIC knows calling object */
%CHOICE, &choicevar;
{
                                       /* "1. Aircraft" */
     %CASE= 49:                        /* "1" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Aircraft", 1;
     %CASE= 65:                        /* "A" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Aircraft", 1;
```

```
                                    /* "2. aircraft Carriers"*/
%CASE= 50:                            /* "2" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Carriers", 1;
%CASE= 67:                            /* "C" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Carriers", 1;

                                    /*"3. Surface combatants"*/
%CASE= 51:                            /* "3" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "SurfComb", 1;
%CASE= 83:                            /* "S" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "SurfComb", 1;

                                    /* "4. sUbmarines" */
%CASE= 52:                            /* "4" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Subs", 1;
%CASE= 85:                            /* "U" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Subs", 1;

                                    /* "5. auXiliary ships" */
%CASE= 53:                            /* "5" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Auxils", 1;
%CASE= 88:                            /* "X" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Auxils", 1;

                                    /* "6. Merchant ships" */
%CASE= 54:                            /* "6" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Merships", 1;
%CASE= 77:                            /* "M" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Merships", 1;

                                    /* "7. Other units" */
%CASE= 55:                            /* "7" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;
%CASE= 79:                            /* "O" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;

                                    /*"8.Return to prev.menu"*/
%CASE= 56:                            /* "8" in ASCII */
```

```
                @clearscreen = 1;
                %EXECUTE, "MainMenu", 1;
        %CASE= 82:                          /* "R" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "MainMenu", 1;

                                          /* "9. Exit GEOTREC" */
        %CASE= 57:                          /* "9" in ASCII */
                %CALL, "COMMON", 4;
        %CASE= 69:                          /* "E" in ASCII */
                %CALL, "COMMON", 4;

        %DEFAULT:                              /* Invalid selection*/
                %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "UnitMenu", 1;
END_OBJECT
```

G. AIRCRAFT.YOB

```
/* 1: Aircraft */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 9/*lt.blue bkgnd*/;
     @clearscreen = 0;
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> AIRCRAFT ",
     "                          ";
%IF, @disdoc = 1;                  /* only if called from above*/
{
    %DISDOC, "Forger_1", "G", 1,1, 0,0, 0/*bkgnd on white*/,
        50,10, 590,300;
    @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @menuwin, 215,25, 2/*black on green*/,
     "     AIRCRAFT     ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 330,220, 10/*black on lt.green*/,
     " Press number or first ",,
     "   capital letter of ",,
     "   desired selection: ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 333,266, 10/*black on lt.green*/,
"                          ",,
" 1. Bombers/strike",,             /* 1 = ASCII 49; B = ASCII 66*/
" 2. Fighters",,                   /* 2 = ASCII 50; F = ASCII 70*/
" 3. Helicopters",,                /* 3 = ASCII 51; H = ASCII 72*/
" 4. MPA/ASW/recon/EW",,           /* 4 = ASCII 52; M = ASCII 77*/
" 5. Other",,                      /* 5 = ASCII 53; O = ASCII 79*/
" 6. Return to prev.menu",,        /* 6 = ASCII 54; R = ASCII 82*/
" 7. Exit GEOTREC",,               /* 7 = ASCII 55; E = ASCII 69*/
"                        ";
%CALL, "COMMON", 5;                /* close all windows */
@callerobj = "Aircraft";/* so GENERIC knows calling object */
%CHOICE, &choicevar;
{
                                   /* "1. Bombers/strike" */
     %CASE= 49:                    /* "1" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "BombStk", 1;
     %CASE= 66:                    /* "B" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "BombStk", 1;
                                   /* "2. Fighters" */
     %CASE= 50:                    /* "2" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
```

```
                %EXECUTE, "Fighters", 1;
        %CASE= 70:                      /* "F" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Fighters", 1;
                                        /* "3. Helicopters" */
        %CASE= 51:                      /* "3" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Helos", 1;
        %CASE= 72:                      /* "H" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "Helos", 1;

                                      /* "4. MPA/ASW/recon/EW" */
        %CASE= 52:                      /* "4" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "MPAASWEW", 1;*/

        %CASE= 77:                      /* "M" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "MPAASWEW", 1;*/

                                        /* "5. Other" */
        %CASE= 53:                      /* "5" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "OthrAcft", 1;*/

        %CASE= 79:                      /* "O" in ASCII */
                %CALL, "COMMON", 6;/* sets vars to = 1 */
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "OthrAcft", 1;*/

                                     /*"6.Return to prev.menu"*/
        %CASE= 54:                      /* "6" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;
        %CASE= 82:                      /* "R" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;

                                        /* "7. Exit GEOTREC" */
        %CASE= 55:                      /* "7" in ASCII */
                %CALL, "COMMON", 4;
        %CASE= 69:                      /* "E" in ASCII */
                %CALL, "COMMON", 4;
        %DEFAULT:                       /* Invalid Selection */
                %CALL, "COMMON", 3;
                @clearscreen = 0;
)/* end CHOICE */
```

```
%EXECUTE, "Aircraft", 1;
END_OBJECT
```

H. BOMBSTK.YOB

```
/* 1: BombStk */
%IF, @clearscreen = 1;          /* only if called from above*/
{
     %CLEARSCREEN, 14/*yellow bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
    " MENU LEVEL:  Main --> Unit --> Aircraft --> "
    "BOMBER/STRIKE ACFT   ";
%IF, @disdoc = 1;               /* only if called from above*/
{
    %DISDOC, "Blackj_1", "G", 0, 4, 0,0, 15/*black on bkgnd*/,
         100,45, 545,285;
    @disdoc = 0;
}/* end IF */
%CALL, "COMMON", 1;                  /* mouse instructions */
%FONT, "System72.fnt";
%MENU, &sortvar, 200, 270, 1/*default choice*/,
     " LIST AIRCRAFT NAMES SORTED BY:  ",
     "DESIGNATOR   ",                    /* choice 1 */
     "NATO CODENAME",                    /* choice 2 */
     "--------------",                   /* separator */
     "[return to previous menu]",        /* choice 4 */
     "[exit GEOTREC]";                   /* choice 5 */
%CHOICE, &sortvar;
{
     %CASE= 1:    %EXECUTE, "BombStk", 2;
     %CASE= 2:    %EXECUTE, "BombStk", 3;
     %CASE= 4:    @clearscreen = 1;
                  %EXECUTE, "Aircraft", 1;
     %CASE= 5:    %CALL, "COMMON", 4;
     %DEFAULT:    %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "BombStk", 1;
END_OBJECT

/* 2: Menu, SORTED BY DESIGNATOR */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 14/*yellow bkgnd*/;
     %CALL, "COMMON", 1;             /* mouse instructions*/
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
    " MENU LEVEL:  Main --> Unit --> Acft --> "
    "Bmb/Stk --> BY DESIGNATOR";
%FONT, "System72.fnt";
```

```
%MENU, &choicevar, 205,125, 5/*default choice*/,
     " BOMBERS/STRIKE AIRCRAFT ",
     " - SORTED BY DESIGNATOR - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "Tu-16  Badger C/G",            /* choice 5 */
     "Tu-22  Blinder   ",            /* choice 6 */
     "Tu-22m Backfire  ",            /* choice 7 */
     "Tu-95  Bear B/C  ",            /* choice 8 */
     "Tu-160 Blackjack ",            /* choice 9 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "BombStk", 2;
%CHOICE, &choicevar;
{
     %CASE= 2:                       /* "[Return to prev.menu]"*/
                %EXECUTE, "BombStk", 1;

     %CASE= 3:                       /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 5:                       /* "Tu-16 Badger C/G" */
                %CALL, "GENERIC", 3;
                @clearscreen = 1;
/*              %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Badger_1", "G", 1, 1, 0,0,
                   1 *color* , 50,45, 620,460;*/

     %CASE= 6:                       /* "Tu-22 Blinder" */
                %CALL, "GENERIC", 3;
                @clearscreen = 1;
/*              %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Blinder1", "G", 1, 1, 0,0,
                   1 *color* , 50,45, 620,460;*/

     %CASE= 7:                       /* "Tu-22m Backfire" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Backf_1", "G", 1, 1, 0,0,
                   1/*color*/, 70,140, 570,315;

     %CASE= 8:                       /* "Tu-95 Bear B/C" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "TU95", "G", 1, 1, 0,0,
                   1/*color*/, 10,90, 630,370;

     %CASE= 9:                       /* "Tu-160 Blackjack" */
```

```
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Blackj_1", "G", 1, 4, 0,0,
                   1/*color*/, 75,50, 540,300;
     %DEFAULT:                          /* Invalid selection */
                %CALL, "COMMON", 3;
                @clearscreen = 0;
}/* end CHOICE */
%EXECUTE, "BombStk", 2;
END_OBJECT

/* 3: Menu, SORTED BY NATO CODENAME */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 14/*yellow bkgnd*/;
     %CALL, "COMMON", 1;                /* mouse instruct.s*/
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
    " MENU LEVEL:  Main --> Unit --> Acft --> Bmb/Stk --> "
    "BY CODENAME  ";
%FONT, "System72.fnt";
%MENU, &choicevar, 205,125, 5/*default choice*/,
     " BOMBERS/STRIKE AIRCRAFT ",
     "- SORTED BY NATO CODENAME -",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "Backfire (Tu-22m) ",           /* choice 5 */
     "Badger C/G (Tu-16)",           /* choice 6 */
     "Bear B/C (Tu-95)  ",           /* choice 7 */
     "Blackjack (Tu-160)",           /* choice 8 */
     "Blinder (Tu-22)   ",           /* choice 9 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "BombStk", 3;
%CHOICE, &choicevar;
{
     %CASE= 2:                     /*"[Return to prev.menu]"*/
                %EXECUTE, "BombStk", 1;

     %CASE= 3:                          /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 5:                        /* "Backfire (Tu-22m)" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Backf_1", "G", 1, 1, 0,0,
                   1/*color*/, 70,140, 570,315;

     %CASE= 6:                       /* "Badger C/G (Tu-16)" */
```

```
                    %CALL, "GENERIC", 3;
                    @clearscreen = 1;
/*                  %CALL, "COMMON", 2;
                    @clearscreen = 1;
                    %PROCESSDOC, "Badger_1", "G", 1, 1, 0,0,
                       1 *color* , 50,45, 620,460;*/

     %CASE= 7:                          /* "Bear B/C (Tu-95)" */
                    %CALL, "COMMON", 2;
                    @clearscreen = 1;
                    %PROCESSDOC, "TU95", "G", 1, 1, 0,0,
                       1/*color*/, 10,90, 630,370;

     %CASE= 8:                         /* "Blackjack (Tu-160)" */
                    %CALL, "COMMON", 2;
                    @clearscreen = 1;
                    %PROCESSDOC, "Blackj_1", "G", 1, 4, 0,0,
                       1/*color*/, 75,50, 540,300;
     %CASE= 9:                           /* "Blinder (Tu-22)" */
                    %CALL, "GENERIC", 3;
                    @clearscreen = 1;
/*                  %CALL, "COMMON", 2;
                    @clearscreen = 1;
                    %PROCESSDOC, "Blinder1", "G", 1, 1, 0,0,
                       1 *color* , 50,45, 620,460;*/
     %DEFAULT:                          /* Invalid Selection */
                    %CALL, "COMMON", 3;
                    @clearscreen = 0;
}/* end CHOICE */
%EXECUTE, "BombStk", 3;
END_OBJECT

/* 4: Backfire text */
@clearscreen = 1;
%PROCESSDOC, "Backfire", "T", 1,1, 0,0, 10/*blk on lt.green*/,
     0,75, 640,300;
END_OBJECT

/* 5: AS-4 on Backfire (underside) */
@clearscreen = 1;
%PROCESSDOC, "Backf_2", "G", 1,1, 0,0, 1/*color*/,
     115,115, 475,350;
END_OBJECT

/* 6: AS-15 text */
@clearscreen = 1;
%PROCESSDOC, "AS15", "T", 1,1, 0,0, 10/*black on lt.green*/,
     75,25, 590,200;
END_OBJECT

/* 7: Blackjack text */
```

```
%DISTEXT, &blackjwin, 15,15,15,
     " This is an artist's conception of a Tu-160 Blackjack.";
%WAIT,,3.0;
%CLOSEWIN, &blackjwin;
END_OBJECT
```

I. FIGHTERS.YOB

```
/* 1: Fighters */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 10/*lt.green bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
  " MENU LEVEL:  Main --> Unit --> Aircraft --> FIGHTERS ",
  "           ";
%IF, @disdoc = 1;
{
    %DISDOC, "Flanker", "G", 1, 2, 0,0, 15/*black on bkgnd*/,
         40,40, 600,320;
    @disdoc = 0;
}/* end IF */
%CALL, "COMMON", 1;              /* mouse instructions */
%FONT, "System72.fnt";
%MENU, &sortvar, 200, 270, 1/*default choice*/,
     " LIST AIRCRAFT NAMES SORTED BY:  ",
     "DESIGNATOR   ",                    /* choice 1 */
     "NATO CODENAME",                    /* choice 2 */
     "---------------",                  /* separator */
     "[return to previous menu]",        /* choice 4 */
     "[exit GEOTREC]";                   /* choice 5 */
%CHOICE, &sortvar;
{
     %CASE= 1:   %EXECUTE, "Fighters", 2;
     %CASE= 2:   %EXECUTE, "Fighters", 3;
     %CASE= 4:   @clearscreen = 1;
                 %EXECUTE, "Aircraft", 1;
     %CASE= 5:   %CALL, "COMMON", 4;
     %DEFAULT:   %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "Fighters", 1;
END_OBJECT

/* 2: Menu, SORTED BY DESIGNATOR */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 10/*lt.green bkgnd*/;
     %CALL, "COMMON", 1;         /* mouse instructions */
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> Acft --> Fighters ",
     "--> BY DESIGNATOR";
%FONT, "System72.fnt";
%MENU, &choicevar, 210,100, 5/*default choice*/,
```

```
      " FIGHTER AIRCRAFT ",
      " - SORTED BY DESIGNATOR - ",
      "[return to previous menu]",    /* choice 2 */
      "[exit GEOTREC]",               /* choice 3 */
      "--------------------",         /* separator */
      "MiG-23 Flogger   ",            /* choice 5 */
      "MiG-25 Foxbat    ",            /* choice 6 */
      "MiG-29 Fulcrum   ",            /* choice 7 */
      "MiG-31 Foxhound  ",            /* choice 8 */
      "Su-20 Fitter     ",            /* choice 9 */
      "Su-24 Fencer     ",            /* choice 10*/
      "Su-25 Frogfoot   ",            /* choice 11*/
      "Su-27 Flanker    ",            /* choice 12*/
      "Yak-38 Forger    .",           /* choice 13*/
      "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Fighters", 2;
%CHOICE, &choicevar;
{
     %CASE= 2:                    /* "[Return to prev.menu]"*/
                %EXECUTE, "Fighters", 1;

     %CASE= 3:                          /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 4:   %CALL, "COMMON", 3;/* Invalid Selection */
                 @clearscreen = 0;

     %CASE= 10:                         /* "Su-24 FENCER" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "SU24", "G", 1, 2, 0,0,
                    1/*color*/, 40,40, 600,320;

     %CASE= 12:                         /* "Su-27 FLANKER" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Flanker", "G", 1, 2, 0,0,
                    1/*color*/, 40,40, 600,320;

     %CASE= 13:                         /* "Yak-38 FORGER" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Forger_1", "G", 1, 1, 0,0,
                        1/*color*/, 40,40, 600,320;

     %CASE= 14:  %CALL, "COMMON", 3;/* Invalid Selection */
                 @clearscreen = 0;

     %DEFAULT:          /* SELECTION NOT YET AVAILABLE */
                %CALL, "GENERIC", 3;
```

```
                    @clearscreen = 1;
}/* end CHOICE */
%EXECUTE, "Fighters", 2;
END_OBJECT

/* 3: Menu, SORTED BY NATO CODENAME */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 10/*lt.green bkgnd*/;
     %CALL, "COMMON", 1;          /* mouse instructions */
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> Acft --> Fighters ",
     "--> BY CODENAME ";
%FONT, "System72.fnt";
%MENU, &choicevar, 210,100, 5/*default choice*/,
     " FIGHTER AIRCRAFT ",
     " - SORTED BY NATO CODENAME - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "FENCER (Su-24)   ",            /* choice 5 */
     "FITTER (Su-20/22)",            /* choice 6 */
     "FLANKER (Su-27)  ",            /* choice 7 */
     "FLOGGER (MiG-23) ",            /* choice 8 */
     "FORGER (Yak-38)  ",            /* choice 9 */
     "FOXBAT (MiG-25)  ",            /* choice 10*/
     "FOXHOUND (MiG-25)",            /* choice 11*/
     "FROGFOOT (Su-25) ",            /* choice 12*/
     "FULCRUM (MiG-29) ",            /* choice 13*/
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Fighters", 3;
%CHOICE, &choicevar;
{
     %CASE= 2:                       /*"[Return to prev.menu]"*/
                %EXECUTE, "Fighters", 1;

     %CASE= 3:                       /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 4:  %CALL, "COMMON", 3;/* Invalid Selection */
                @clearscreen = 0;

     %CASE= 5:                       /* "FENCER (Su-24)" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "SU24", "G", 1, 2, 0,0,
                    1/*color*/, 40,40, 600,320;
```

```
      %CASE= 7:                          /* "FLANKER (Su-27)" */
                  %CALL, "COMMON", 2;
                  @clearscreen = 1;
                  %PROCESSDOC, "Flanker", "G", 1, 2, 0,0,
                      1/*color*/, 40,40, 600,320;

      %CASE= 9:                           /* "FORGER (Yak-38)" */
                  %CALL, "COMMON", 2;
                  @clearscreen = 1;
                  %PROCESSDOC, "Forger_1", "G", 1, 1, 0,0,
                      1/*color*/, 70,140, 570,315;

      %CASE= 14:  %CALL, "COMMON", 3;/* Invalid Selection */
                  @clearscreen = 0;

      %DEFAULT:           /* SELECTION NOT YET AVAILABLE */
                  %CALL, "GENERIC", 3;
                  @clearscreen = 1;
}/* end CHOICE */
%EXECUTE, "Fighters", 3;
END_OBJECT
```

J. HELOS.YOB

```
/* 1: Helos */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 13/*lt.magenta bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
 " MENU LEVEL:  Main --> Unit --> Aircraft -->   ",
 " HELICOPTERS          ";
%IF, @disdoc = 1;
{
    %DISDOC, "Helix_1", "G", 1, 2, 0,0, 15/*black on bkgnd*/,
         0,0, 640,480;
    @disdoc = 0;
}/* end IF */
%CALL, "COMMON", 1;                    /* mouse instructions */
%FONT, "System72.fnt";
%MENU, &sortvar, 50,50, 1/*default choice*/,
     " LIST AIRCRAFT NAMES SORTED BY:  ",
     "DESIGNATOR   ",                   /* choice 1 */
     "NATO CODENAME",                   /* choice 2 */
     "---------------",                 /* separator */
     "[return to previous menu]",       /* choice 4 */
     "[exit GEOTREC]";                  /* choice 5 */
%CHOICE, &sortvar;
{
     %CASE= 1:   %EXECUTE, "Helos", 2;
     %CASE= 2:   %EXECUTE, "Helos", 3;
     %CASE= 4:   @clearscreen = 1;
                 %EXECUTE, "Aircraft", 1;
     %CASE= 5:   %CALL, "COMMON", 4;
     %DEFAULT:   %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "Helos", 1;
END_OBJECT

/* 2: Menu, SORTED BY DESIGNATOR */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 13/*lt.magenta bkgnd*/;
     %CALL, "COMMON", 1;               /* mouse instructions */
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
 " MENU LEVEL:  Main --> Unit --> Acft --> ",
 " Helos --> BY DESIGNATOR ";
%FONT, "System72.fnt";
%MENU, &choicevar, 210,100, 5/*default choice*/,
```

```
     "     HELICOPTERS     ",
     " - SORTED BY DESIGNATOR - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "Ka-25 Hormone     ",           /* choice 5 */
     "Ka-27 Helix       ",           /* choice 6 */
     "Mi-8 Hip          ",           /* choice 7 */
     "Mi-14 Haze        ",           /* choice 8 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Helos", 2;
%CHOICE, &choicevar;
{
     %CASE= 2:                  /* "[Return to prev.menu]" */
               %EXECUTE, "Helos", 1;

     %CASE= 3:                        /* "[Exit GEOTREC]" */
               %CALL, "COMMON", 4;
               @clearscreen = 0;

     %CASE= 4:   %CALL, "COMMON", 3;/* Invalid Selection */
               @clearscreen = 0;

     %CASE= 5:                        /* "Ka-25 Hormone" */
               %CALL, "COMMON", 2;
               @clearscreen = 1;
               %PROCESSDOC, "VII_Hor2", "G", 1,1, 0,0,
                   1/*color*/, 30,40, 610,290;

     %CASE= 6:                       /* "Ka-27 Helix" */
               %CALL, "COMMON", 2;
               @clearscreen = 1;
               %PROCESSDOC, "HelixBig", "G", 1, 1, 0,0,
                   1/*color*/, 10,30, 630,440;

     %CASE= 9:   %CALL, "COMMON", 3;/* Invalid Selection */
               @clearscreen = 0;

     %DEFAULT:             /* SELECTION NOT YET AVAILABLE */
               %CALL, "GENERIC", 3;
               @clearscreen = 1;
}/* end CHOICE */
%EXECUTE, "Helos", 2;
END_OBJECT

/* 3: Menu, SORTED BY NATO CODENAME */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 13/*lt.magenta bkgnd*/;
     %CALL, "COMMON", 1;             /* mouse instructions */
}/* end IF */
```

```
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs*/, 8/*dk.grey bkgnd*/,
 " MENU LEVEL:  Main --> Unit --> Acft --> ",
 " Helos --> BY CODENAME ";
%FONT, "System72.fnt";
%MENU, &choicevar, 210,100, 5/*default choice*/,
     "     HELICOPTERS     ",
     " - SORTED BY NATO CODENAME - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "HAZE (Mi-14)      ",           /* choice 5 */
     "HELIX (Ka-27)     ",           /* choice 6 */
     "HIP (Mi-8)        ",           /* choice 7 */
     "HORMONE (Ka-25)   ",           /* choice 8 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Helos", 3;
%CHOICE, &choicevar;
{
     %CASE= 2:                    /* "[Return to prev.menu]" */
                %EXECUTE, "Helos", 1;

     %CASE= 3:                     /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 4:  %CALL, "COMMON", 3;/* Invalid Selection */
                @clearscreen = 0;

     %CASE= 6:                    /* "Ka-27 Helix" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "HelixBig", "G", 1, 1, 0,0,
                    1/*color*/, 10,30, 630,440;

     %CASE= 8:                     /* "Ka-25 Hormone" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "VII_Hor2", "G", 1,1, 0,0,
                    1/*color*/, 30,40, 610,290;

     %CASE= 9:  %CALL, "COMMON", 3;/* Invalid Selection */
                @clearscreen = 0;

     %DEFAULT:          /* SELECTION NOT YET AVAILABLE */
                %CALL, "GENERIC", 3;
                @clearscreen = 1;
}/* end CHOICE */
%EXECUTE, "Helos", 3;
END_OBJECT
```

K. CARRIERS.YOB

```
/* 1: Carriers */
%IF, @clearscreen = 1;       /* set by UnitMenu object */
{
     %CLEARSCREEN, 11/*lt.azure bkgnd*/;
     @clearscreen = 0;
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MAIN MENU:  Main --> Unit --> AIRCRAFT CARRIERS",
     "                    ";
%IF, @disdoc = 1;           /* set by UnitMenu object */
{
    %DISDOC, "Tbilisi1", "G", 0, 1, 0,0, 15/*black on bkgnd*/,
        75,50, 550,325;
    @disdoc = 0;
}/* end IF */
%CALL, "COMMON", 1;               /* mouse instructions */
%FONT, "System72.fnt";
%MENU, &sortvar, 250, 300, 1/*default choice*/,
     " LIST AIRCRAFT CARRIERS BY:  ",
     "TYPE (CV, CVHG, etc.)",           /* choice 1 */
     "CLASS NAME",                      /* choice 2 */
     "-------------------------",       /* separator */
     "[return to previous menu]",       /* choice 4 */
     "[exit GEOTREC]";                  /* choice 5 */
%CHOICE, &sortvar;
{
     %CASE= 1:   %EXECUTE, "Carriers", 2;
     %CASE= 2:   %EXECUTE, "Carriers", 3;
     %CASE= 4:   @clearscreen = 1;
                 %EXECUTE, "UnitMenu", 1;
     %CASE= 5:   %CALL, "COMMON", 4;
                 %EXECUTE, "Carriers", 1;
     %DEFAULT:   %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "Carriers", 1;
END_OBJECT

/* 2: Menu, SORTED BY TYPE */
%IF, @clearscreen = 1;/*screen cleared except 1st time thru*/
{
     %CLEARSCREEN, 11/*lt.azure bkgnd*/;
     %CALL, "COMMON", 1;            /* mouse instructions */
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
 " MENU LEVEL:  Main --> Unit --> Carriers --> ",
 " SORTED BY TYPE          ";
%FONT, "System72.fnt";
```

```
%MENU, &choicevar, 225,100, 5/*default choice*/,
     " AIRCRAFT CARRIERS ",
     " - SORTED BY TYPE - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "CHG (MOSKVA)     ",            /* choice 5 */
     "CVHG (KIEV)      ",            /* choice 6 */
     "CVxx (TBILISI)   ",            /* choice 7 */
     "CVyy (ULYANOVSK)",             /* choice 8 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Carriers", 2;
%CHOICE, &choicevar;
{
     %CASE= 2:                  /*"[Return to prev.menu]"*/
                %EXECUTE, "Carriers", 1;

     %CASE= 3:                        /* "[Exit GEOTREC]" */
                %CALL, "COMMON", 4;
                @clearscreen = 0;

     %CASE= 5:                        /* "CHG (MOSKVA)" */
                %CALL, "GENERIC", 3;
                @clearscreen = 1;

     %CASE= 6:                        /* "CVHG (KIEV)" */
                %CALL, "COMMON", 2;
                @clearscreen = 1;
                %PROCESSDOC, "Baku", "G", 1,1, -20,0,
                1/*color*/, 150,90, 515,390;

     %CASE= 7:                        /* "CVxx (TBILISI)" */
                %CALL, "COMMON", 2;
                @clearscreen = 1; /*so screen'll clear */
                %PROCESSDOC, "Tbilisi1", "G", 1, 1, 0,0,
                   2/*blk on wt*/, 50,50, 550,325;

     %CASE= 8:                       /* "CVyy (ULYANOVSK)" */
                %CALL, "GENERIC", 3;
                @clearscreen = 1;
     %DEFAULT:                        /* Invalid selection */
                %CALL, "COMMON", 3;
                @clearscreen = 0;
}/* end CHOICE */
%EXECUTE, "Carriers", 2;
END_OBJECT

/* 3: Menu, SORTED BY CLASS NAME */
%IF, @clearscreen = 1;/*screen cleared except 1st time thru*/
{
     %CLEARSCREEN, 11/*lt.azure bkgnd*/;
```

```
     %CALL, "COMMON", 1;              /* mouse instructions */
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> Carriers --> ",
     "SORTED BY CLASS NAME ";
%FONT, "System72.fnt";
%MENU, &choicevar, 225,100, 5/*default choice*/,
     " AIRCRAFT CARRIERS ",
     " - SORTED BY CLASS NAME - ",
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     "KIEV CVHG      ",              /* choice 5 */
     "MOSKVA CHG     ",              /* choice 6 */
     "TBILISI CVxx   ",              /* choice 7 */
     "ULYANOVSK CVyy",               /* choice 8 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "Carriers", 3;
%CHOICE, &choicevar;
{
     %CASE= 2:                       /* previous menu */
               %EXECUTE, "Carriers", 1;

     %CASE= 3:                       /* "[Exit GEOTREC]" */
               %CALL, "COMMON", 4;
               @clearscreen = 0;

     %CASE= 5:                       /* "KIEV CVHG" */
               %CALL, "COMMON", 2;
               @clearscreen = 1;
               %PROCESSDOC, "Baku", "G", 1,1, -20,0,
               1/*color*/, 150,90, 515,390;

     %CASE= 6:                       /* "MOSKVA CHG" */
               %CALL, "GENERIC", 3;
               @clearscreen = 1;

     %CASE= 7:                       /* "TBILISI CVxx" */
               %CALL, "COMMON", 2;
               @clearscreen = 1;/* so screen'll clear*/
               %PROCESSDOC, "Tbilisi1", "G", 1, 1, 0,0,
                  2/*blk on wt*/, 50,50, 550,325;

     %CASE= 8:                       /* "ULYANOVSK CVyy" */
               %CALL, "GENERIC", 3;
               @clearscreen = 1;

     %DEFAULT:                       /* Invalid selection */
               %CALL, "COMMON", 3;
               @clearscreen = 0;
```

```
}/* end CHOICE */
%EXECUTE, "Carriers", 3;
END_OBJECT
```

L. SURFCOMB.YOB

```
/* 1: SurfComb */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 15;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> SURFACE COMBATANTS ",
     "                ";
%IF, @disdoc = 1;
{
    %DISDOC, "Slava_2", "G", 0, 1, 0,-10, 15/*blk on bkgnd*/,
        30,20, 610,375;
     @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @menuwin, 355,60, 13/*black on lt.magenta*/,
     " SURFACE COMBATANTS ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 105,215, 11/*black on lt.azure*/,
     " Press number or first ",,
     "   capital letter of ",,
     "   desired selection: ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 108,261, 11/*black on lt.azure*/,
   "                          ",,
   " 1. Major",,              /* 1 = ASCII 49; M = ASCII 77 */
   " 2. mInor",,              /* 2 = ASCII 50; I = ASCII 73 */
   " 3. Patrol craft",,       /* 3 = ASCII 51; P = ASCII 80 */
   " 4. mineSweepers",,       /* 4 = ASCII 52; S = ASCII 83 */
   " 5. Amphibs/landing crft",,/*5= ASCII 53; A = ASCII 65 */
   " 6. Return to prev.menu",,/*6 = ASCII 54; R = ASCII 82 */
   " 7. Exit GEOTREC",,       /* 7 = ASCII 55; E = ASCII 69 */
   "                          ";
%CALL, "COMMON", 5;
@callerobj = "SurfComb";
%CHOICE, &choicevar;
{
                                    /* "6. Return to prev.menu"*/
     %CASE= 54:                        /* "6" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;
     %CASE= 82:                        /* "R" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;

                                       /* "7. Exit GEOTREC" */
     %CASE= 55:                        /* "7" in ASCII */
```

```
                %CALL, "COMMON", 4;
        %CASE= 69:                      /* "E" in ASCII */
                %CALL, "COMMON", 4;

        %DEFAULT:                   /* Selection not yet available*/
                %CALL, "COMMON", 6;
                %CALL, "GENERIC", 1;
}/* end CHOICE */
%EXECUTE, "SurfComb", 1;
END_OBJECT
```

M. TARANTUL.YOB

```
/* 1: TARANTUL (if user clicks on HULL) */
@returnobj = "TARANTUL";
@returnnum = 8;
%FUNCTIONKEY, "ESC", @returnobj, @returnnum;
%FONT, "System24.fnt";
%PROCESSDOC, "TARANTUL", "T", 1,1, 0,0,
     11/*black on lt.azure*/, 35,365, 605,450;
END_OBJECT

/* 2: if user clicks on BANDSTAND radar */
%FONT, "System24.fnt";
%DISTEXT, &textwin, 300,100, 10/*black on lt.green*/,
     " This is the BANDSTAND radar. ";
%WAIT,,1.0;
%DISTEXTSTOP, &key, 125,445, 14/*black on yellow*/,
     " (Press M for More info on your selection, ",,
     "  any other key to make another selection) ";
%CLOSEWIN, &textwin;
%CHOICE, &key;
{
     %CASE= 77:                          /* "M" in ASCII */
                %CALL, "GENERIC", 3;
                %EXECUTE, "TARANTUL", 8;
     %DEFAULT:  %EXECUTE, "TARANTUL", 7;
}/* end CHOICE */
END_OBJECT

/* 3: if user clicks on SS-N-22 launchers */
%FONT, "System24.fnt";
%DISTEXT, &textwin, 225,300, 10/*black on lt.green*/,
     " This is an SS-N-22 launcher. ";
%WAIT,,1.0;
%DISTEXTSTOP, &key, 125,445, 14/*black on yellow*/,
     " (Press M for More info on your selection, ",,
     "  any other key to make another selection) ";
%CLOSEWIN, &textwin;
%CHOICE, &key;
{
     %CASE= 77:                          /* "M" in ASCII */
                %FUNCTIONKEY, "ESC", "TARANTUL", 8;
                %EXECUTE, "TARANTUL", 9;
     %DEFAULT:  %EXECUTE, "TARANTUL", 7;
}/* end CHOICE */
END_OBJECT

/* 4: if user clicks on 76mm gun */
%FONT, "System24.fnt";
%DISTEXT, &textwin, 400,150, 10/*black on lt.green*/,
     " This is a 76mm dual-purpose naval gun. ";
```

```
%WAIT,,1.0;
%DISTEXTSTOP, &key, 125,445, 14/*black on yellow*/,
     " (Press M for More info on your selection, ",,
     "  any other key to make another selection) ";
%CLOSEWIN, &textwin;
%CHOICE, &key;
{
     %CASE= 77:                          /* "M" in ASCII */
                %FUNCTIONKEY, "ESC", "TARANTUL", 8;
                %EXECUTE, "TARANTUL", 10;
     %DEFAULT:  %EXECUTE, "TARANTUL", 7;
}/* end CHOICE */
END_OBJECT

/* 5: if user clicks on BASS TILT radar */
%FONT, "System24.fnt";
%DISTEXT, &textwin, 225,100, 10/*black on lt.green*/,
     " This is the BASS TILT radar. ";
%WAIT,,1.0;
%DISTEXTSTOP, &key, 125,445, 14/*black on yellow*/,
     " (Press M for More info on your selection, ",,
     "  any other key to make another selection) ";
%CLOSEWIN, &textwin;
%CHOICE, &key;
{
     %CASE= 77:                          /* "M" in ASCII */
                %CALL, "GENERIC", 3;
                %EXECUTE, "TARANTUL", 8;
     %DEFAULT:  %EXECUTE, "TARANTUL", 7;
}/* end CHOICE */
END_OBJECT

/* 6: if user clicks on ADMG-630 */
%FONT, "System24.fnt";
%DISTEXT, &textwin, 30,150, 10/*black on lt.green*/,
     " This is an ADMG-630 gatling gun. ";
%WAIT,,1.0;
%DISTEXTSTOP, &key, 125,445, 14/*black on yellow*/,
     " (Press M for More info on your selection, ",,
     "  any other key to make another selection) ";
%CLOSEWIN, &textwin;
%CHOICE, &key;
{
     %CASE= 77:                          /* "M" in ASCII */
                @returnobj = "TARANTUL";
                @returnnum = 8;
                %EXECUTE, "TARANTUL", 11;
     %DEFAULT:  %EXECUTE, "TARANTUL", 7;
}/* end CHOICE */
END_OBJECT
```

```
/* 7: to re-PROCESSDOC("G") Tarantul w/out redraw */
%FUNCTIONKEY, "ESC", @callerobj, @callernum;
%PROCESSDOC, "TARANTUL", "G";
END_OBJECT

/* 8: if user ESCs from PROCESSDOC("T") */
%FUNCTIONKEY, "ESC", @callerobj, @callernum;
%CALL, "COMMON", 2;
%PROCESSDOC, "TARANTUL", "G", 1,1, 0,0, 1/*color*/,
     20,25, 620,455;
END_OBJECT

/* 9: call PROCESSDOC("T") SS-N-22 */
             /* This is broken out from "TARANTUL", 3   *
              * so that it can also be called by a user *
              * 'click' from PROCESSDOC, "TARANTUL", "G"*
              */
%EXECUTE, "SSN22", 1;
END_OBJECT

/* 10: call PROCESSDOC("T") 76mm*/
%CALL, "GENERIC", 3;
%EXECUTE, "TARANTUL", 8;
END_OBJECT

/* 11: call PROCESSDOC ADMG-630 */
             /* This is broken out from "TARANTUL", 6   *
              * so that it can also be called by a user *
              * 'click' from PROCESSDOC, "TARANTUL", "G"*
              */
%EXECUTE, "ADMG630", 1;
END_OBJECT
```

N. SUBS.YOB

```
/* 1: Subs */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 9/*lt.blue bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> SUBMARINES ",
     "                    ";
%IF, @disdoc = 1;
{
     %DISDOC, "VII_Hor2", "G", 0, 1, 0,0, 0/*bkgnd on white*/,
        10,10, 630,250;
     @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @menuwin, 200,75, 5/*white on red*/,
     "     SUBMARINES     ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 200,235, 15/*black on white*/,
     " Press number or first ",,
     "   capital letter of ",,
     "   desired selection: ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,281, 15/*black on white*/,
   "                      ",,
   " 1. Attack (SSN/SS)",,   /* 1 = ASCII 49; A = ASCII 65 */
   " 2. Boomers (SSBN/SSB)",,/* 2 = ASCII 50; B = ASCII 66 */
   " 3. Shooters (SSGN/SSG) ",,/*3= ASCII 51; S = ASCII 83 */
   " 4. Other submarines",,  /* 4 = ASCII 52; O = ASCII 79 */
   " 5. Return to prev.menu",,/*5 = ASCII 53; R = ASCII 82 */
   " 6. Exit GEOTREC",,      /* 6 = ASCII 54; E = ASCII 69 */
   "                      ";
%CALL, "COMMON", 5;
@callerobj = "Subs";
%CHOICE, &choicevar;
{                                  /* "5. Return to prev.menu"*/
     %CASE= 53:                         /* "5" in ASCII */
               @clearscreen = 1;
               %EXECUTE, "UnitMenu", 1;
     %CASE= 82:                         /* "R" in ASCII */
               @clearscreen = 1;
               %EXECUTE, "UnitMenu", 1;

                                        /* "6. Exit GEOTREC" */
     %CASE= 54:                         /* "6" in ASCII */
               %CALL, "COMMON", 4;
     %CASE= 69:                         /* "E" in ASCII */
```

```
                %CALL, "COMMON", 4;

     %DEFAULT:                  /* selection not yet available*/
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
}/* end CHOICE */
%EXECUTE, "Subs", 1;
END_OBJECT
```

```
0.  AUXILS.YOB
/* 1: Auxils */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 7/*lt.grey bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> AUXILIARIES ",
     "                          ";
%IF, @disdoc = 1;
{
     %DISDOC, "Balzaam", "G", 0, 1, 0,0, 15/*black on bkgnd*/,
        85,10, 530,250;
     @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @menuwin, 200,125, 14/*black on yellow*/,
     "     AUXILIARIES     ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 200,235, 13/*black on lt.magenta*/,
     " Press number or first ",,
     "   capital letter of ",,
     "   desired selection: ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 205,281, 13/*black on lt.magenta*/,
   "                            ",,
   " 1. AGIs/research",,      /* 1 = ASCII 49; A = ASCII 65 */
   " 2. Fleet service",,      /* 2 = ASCII 50; F = ASCII 70 */
   " 3. Space support/C3",,   /* 3 = ASCII 51; S = ASCII 83 */
   " 4. Other auxiliaries",, /* 4 = ASCII 52; O = ASCII 79 */
   " 5. Return to prev.menu",,/*5 = ASCII 53; R = ASCII 82 */
   " 6. Exit GEOTREC",,       /* 6 = ASCII 54; E = ASCII 69 */
   "                          ";
%CALL, "COMMON", 5;
@callerobj = "Auxils";
%CHOICE, &choicevar;
{                                /* "5. Return to prev.menu" */
     %CASE= 53:                        /* "5" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;
     %CASE= 82:                        /* "R" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "UnitMenu", 1;

                                       /* "6. Exit GEOTREC" */
     %CASE= 54:                        /* "6" in ASCII */
                %CALL, "COMMON", 4;
     %CASE= 69:                        /* "E" in ASCII */
```

```
                %CALL, "COMMON", 4;

     %DEFAULT:                  /* selection not yet available*/
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
}/* end CHOICE */
%EXECUTE, "Auxils", 1;
END_OBJECT
```

P. MERSHIPS.YOB

```
/* 1: Merships */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 12/*lt.red bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> Unit --> MERCHANT SHIPS ",
     "                    ";
%IF, @disdoc = 1;
{
     @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @menuwin, 203,125, 0/*white on black*/,
     "   MERCHANT SHIPS    ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 200,180, 9/*black on lt.blue*/,
     " Press number or first ",,
     "   capital letter of ",,
     "   desired selection: ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 203,226, 9/*black on lt.blue*/,
     "                          ",,
     " 1. type 1",,              /* 1 = ASCII 49 */
     " 2. type 2",,              /* 2 = ASCII 50 */
     " 3. type 3",,              /* 3 = ASCII 51 */
     " 4. type 4",,              /* 4 = ASCII 52 */
     " 5. Other merships",,     /* 5 = ASCII 53; O = ASCII 79*/
     " 6. Return to prev.menu",,/*6 = ASCII 54; R = ASCII 82*/
     " 7. Exit GEOTREC",,       /* 7 = ASCII 55; E = ASCII 69*/
     "                          ";
%CALL, "COMMON", 5;
@callerobj = "Merships";
%CHOICE, &choicevar;
{
                                  /* "6. Return to prev.menu"*/
     %CASE= 54:                      /* "6" in ASCII */
               @clearscreen = 1;
               %EXECUTE, "UnitMenu", 1;
     %CASE= 82:                      /* "R" in ASCII */
               @clearscreen = 1;
               %EXECUTE, "UnitMenu", 1;

                                     /* "7. Exit GEOTREC" */
     %CASE= 55:                      /* "7" in ASCII */
               %CALL, "COMMON", 4;
     %CASE= 69:                      /* "E" in ASCII */
```

```
                %CALL, "COMMON", 4;

     %DEFAULT:                  /* selection not yet available */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
}/* end CHOICE */
%EXECUTE, "Merships", 1;
END_OBJECT
```

Q. WEAPMENU.YOB

```
/* 1: WeapMenu */
%IF, @clearscreen = 1;
{
     %CLEARSCREEN, 3/*azure bkgnd*/;
     @clearscreen = 0;
}/* end IF */
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> WEAPONS ",
     "                              ";
%IF, @disdoc = 1;
{
     %DISDOC, "ADMG630", "G", 1,1, 0,-10, 15/*blk on bkg*/,
         10,30, 300,235;
     %DISDOC, "SSN12", "G", 1,1, 0,0, 15/*black on bkgnd*/,
         300,90, 630,380;
     @disdoc = 0;
}/* end IF */
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 355,40, 8/*white on dk.grey*/,
     "     WEAPONS MENU     ";
%FONT, "System24.fnt";
%DISTEXT, @presswin, 106,205, 7/*black on lt.grey*/,
     " Press number or first ",,
     "   capital letter of   ",,
     "   desired selection:  ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 105,251, 7/*black on lt.grey*/,
   "                              ",,
   " 1. Missiles",,          /* 1 = ASCII 49; M = ASCII 77 */
   " 2. Naval guns",,        /* 2 = ASCII 50; N = ASCII 78 */
   " 3. SLBMs",,             /* 3 = ASCII 51; S = ASCII 83 */
   " 4. Torpedoes",,         /* 4 = ASCII 52; T = ASCII 84 */
   " 5. Other weapons",,     /* 5 = ASCII 53; O = ASCII 79 */
   " 6. Return to prev.menu",,/*6 = ASCII 54; R = ASCII 82 */
   " 7. Exit GEOTREC",,      /* 7 = ASCII 55; E = ASCII 69 */
   "                            ";
%CALL, "COMMON", 5;                    /* close windows */
@callerobj = "WeapMenu";/*so GENERIC knows obj that called*/
%CHOICE, &choicevar;
{
                                       /* "1. Missiles" */
     %CASE= 49:                        /* "1" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "Missiles", 1;*/
     %CASE= 77:                        /* "M" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
```

```
/*                %EXECUTE, "Missiles", 1;*/

                                        /* "2. Naval guns" */
     %CASE= 50:                         /* "2" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "NavGuns", 1;*/
     %CASE= 78:                         /* "N" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "NavGuns", 1;*/

                                        /* "3. SLBMs" */
     %CASE= 51:                         /* "3" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "SLBMs", 1;*/
     %CASE= 83:                         /* "S" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "SLBMs", 1;*/

                                        /* "4. Torpedos" */
     %CASE= 52:                         /* "4" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "Torpedos", 1;*/
     %CASE= 84:                         /* "T" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "Torpedos", 1;*/

                                        /* "5. Other weapons" */
     %CASE= 53:                         /* "5" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "OthrWeap", 1;*/
     %CASE= 79:                         /* "O" in ASCII */
                %CALL, "COMMON", 6;
                %EXECUTE, "GENERIC", 1;
/*              %EXECUTE, "OthrWeap", 1;*/

                                    /*"6.Return to prev.menu"*/
     %CASE= 54:                         /* "6" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "MainMenu", 1;
     %CASE= 82:                         /* "R" in ASCII */
                @clearscreen = 1;
                %EXECUTE, "MainMenu", 1;

                                        /* "7. Exit GEOTREC" */
```

```
        %CASE= 55:                          /* "7" in ASCII */
                        %CALL, "COMMON", 4;
        %CASE= 69:                          /* "E" in ASCII */
                        %CALL, "COMMON", 4;

        %DEFAULT:                           /* Invalid selection */
                        %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "WeapMenu", 1;
END_OBJECT
```

R. ADMG630.YOB

```
/* 1: ADMG-630 */
%OPENWIN, @winvar, 125,145, 425,390, 5/*magenta bkgnd*/;
%FONT, "System24.fnt";
%WRITE, 134,370, 15/*white ltrs*/, 5/*magenta bkgnd*/,
     " ADMG-630 (air defense machine ",,
     "   gun with six Mom barrels)  ";
%FUNCTIONKEY, "ESC", "ADMG630", 2;
%FONT, "System24.fnt";
%PROCESSDOC, "ADMG630", "G", 1,1, 0,-10,
    1/*color*/, 130,150, 420,355;
END_OBJECT

/* 2: ESC */
%CLOSEWIN, @winvar;
%EXECUTE, @returnobj, @returnnum;
END_OBJECT
```

S. SSN22.YOB

```
/* 1: SS-N-22 */
%FONT, "System24.fnt";
%PROCESSDOC, "SSN22", "T", 1,1, 0,0,
    13/*black on lt.magenta*/,
    80,380, 560,450;
END_OBJECT
```

T. QUIZZER.YOB

```
/* 1: Welcome to Quizzer */
@usedonebefore = 0;              /* used in "Quizzer", 2 */
@usedtwobefore = 0;              /*  "   "      "      " */
%CLEARSCREEN, 11/*lt.azure bkgnd*/;
%FONT, "System96.fnt";
%DISTEXT, @titlewin, 114,100, 1/*white on blue*/,
     " WELCOME TO THE RECOGNITION QUIZZER !!! ";
%FONT, "System72.fnt";
%DISTEXT, @presswin, 145,170, 9/*white on lt.blue*/,
     " The Quizzer will randomly display threat ",,
     " units, weapons systems or sensors -- your ",,
     " task is to identify the image you see. ";
%FONT, "System72.fnt";
%DISTEXTSTOP, &choicevar, 223,255, 9/*white on lt.blue*/,
     "     PLEASE PRESS:    ",,
     " ~~~~~~~~~~~~~~~~~~~ ",,
     " - S to Start quizzer ",,  /* S = ASCII 83 */
     " - R to Return to     ",,  /* R = ASCII 82 */
     "       Main Menu      ",,
     " - E to Exit GEOTREC  ",,  /* E = ASCII 69 */
     "                      ";
%CALL, "COMMON", 5;              /* close windows */
@callerobj = "Quizzer";
@callernum = 1;
%CHOICE, &choicevar;
{
     %CASE= 83:                  /* "S" in ASCII */
                /* only initialize image variables if *
                 * Quizzer menu called by Main Menu   *
                 */
                %IF, @quizcalledbymain = 1;
                {
                    %CALL, "Quizzer", 5;/*init'lize var.s*/
                    @quizcalledbymain = 0;
                }/* end IF */
                %EXECUTE, "Quizzer", 2;/* start Quizzer */
     %CASE= 82: %CALL, "COMMON", 6;
                %EXECUTE, "MainMenu", 1;
     %CASE= 69: %CALL, "COMMON", 4;
     %DEFAULT:  %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "Quizzer", 1;
END_OBJECT

/* 2: Start Quizzer */
%CLEARSCREEN, 11/*lt.azure bkgnd*/;
                        /* RND(0) returns a whole number *
                         * btwn 1 - 32767; modulo divides*
                         * that set into blocks -- the   *
```

```
                              * number of blocks must be >=   *
                              * the number of @imageXXnames.  *
                              */
&modulo = 2048;/*to divide set of random no.into 16 blocks*/
@rndvar = ROUND(RND(0)/&modulo);
                             /* To ensure the same image is   *
                              * not used twice in a row.      *
                              */
%IF, @rndvar = @usedonebefore;
{
     %EXECUTE, "Quizzer", 2;
}/* end IF */
                             /* To ensure same image is not   *
                              * repeated within two attempts. *
                              */
%IF, @rndvar = @usedtwobefore;
{
     %EXECUTE, "Quizzer", 2;
}/* end IF */
@usedtwobefore = @usedonebefore;
@usedonebefore = @rndvar;
%CHOICE, @rndvar;
{
     %CASE= 1: @name = @01name; @image = @01image; @X = @01X;
               @Y = @01Y; @Xmin = @01Xmin; @Ymin = @01Ymin;
               @Xmax = @01Xmax; @Ymax = @01Ymax;
     %CASE= 2: @name = @02name; @image = @02image; @X = @02X;
               @Y = @02Y; @Xmin = @02Xmin; @Ymin = @02Ymin;
               @Xmax = @02Xmax; @Ymax = @02Ymax;
     %CASE= 3: @name = @03name; @image = @03image; @X = @03X;
               @Y = @03Y; @Xmin = @03Xmin; @Ymin = @03Ymin;
               @Xmax = @03Xmax; @Ymax = @03Ymax;
     %CASE= 4: @name = @04name; @image = @04image; @X = @04X;
               @Y = @04Y; @Xmin = @04Xmin; @Ymin = @04Ymin;
               @Xmax = @04Xmax; @Ymax = @04Ymax;
     %CASE= 5: @name = @05name; @image = @05image; @X = @05X;
               @Y = @05Y; @Xmin = @05Xmin; @Ymin = @05Ymin;
               @Xmax = @05Xmax; @Ymax = @05Ymax;
     %CASE= 6: @name = @06name; @image = @06image; @X = @06X;
               @Y = @06Y; @Xmin = @06Xmin; @Ymin = @06Ymin;
               @Xmax = @06Xmax; @Ymax = @06Ymax;
     %CASE= 7: @name = @07name; @image = @07image; @X = @07X;
               @Y = @07Y; @Xmin = @07Xmin; @Ymin = @07Ymin;
               @Xmax = @07Xmax; @Ymax = @07Ymax;
     %CASE= 8: @name = @08name; @image = @08image; @X = @08X;
               @Y = @08Y; @Xmin = @08Xmin; @Ymin = @08Ymin;
               @Xmax = @08Xmax; @Ymax = @08Ymax;
     %CASE= 9: @name = @09name; @image = @09image; @X = @09X;
               @Y = @09Y; @Xmin = @09Xmin; @Ymin = @09Ymin;
               @Xmax = @09Xmax; @Ymax = @09Ymax;
     %CASE=10: @name = @10name; @image = @10image; @X = @10X;
```

```
                @Y = @10Y; @Xmin = @10Xmin; @Ymin = @10Ymin;
                @Xmax = @10Xmax; @Ymax = @10Ymax;
     %CASE=11: @name = @11name; @image = @11image; @X = @11X;
                @Y = @11Y; @Xmin = @11Xmin; @Ymin = @11Ymin;
                @Xmax = @11Xmax; @Ymax = @11Ymax;
     %CASE=12: @name = @12name; @image = @12image; @X = @12X;
                @Y = @12Y; @Xmin = @12Xmin; @Ymin = @12Ymin;
                @Xmax = @12Xmax; @Ymax = @12Ymax;
     %CASE=13: @name = @13name; @image = @13image; @X = @13X;
                @Y = @13Y; @Xmin = @13Xmin; @Ymin = @13Ymin;
                @Xmax = @13Xmax; @Ymax = @13Ymax;
     %CASE=14: @name = @14name; @image = @14image; @X = @14X;
                @Y = @14Y; @Xmin = @14Xmin; @Ymin = @14Ymin;
                @Xmax = @14Xmax; @Ymax = @14Ymax;
     %CASE=15: @name = @15name; @image = @15image; @X = @15X;
                @Y = @15Y; @Xmin = @15Xmin; @Ymin = @15Ymin;
                @Xmax = @15Xmax; @Ymax = @15Ymax;
     %CASE=16: @name = @16name; @image = @16image; @X = @16X;
                @Y = @16Y; @Xmin = @16Xmin; @Ymin = @16Ymin;
                @Xmax = @16Xmax; @Ymax = @16Ymax;
     %DEFAULT: %EXECUTE, "Quizzer", 2;/*get another rndvar*/
}/* end CHOICE */
%DISDOC, @image, "G", 1,1, @X,@Y, 1/*color*/,
     @Xmin,@Ymin, @Xmax, @Ymax;
%EXECUTE, "Quizzer", 3;
END_OBJECT

/* 3: Quiz Answer */
%CALL, "COMMON", 1;                    /* mouse instructions */
&wronganswer = 0;
&whilevar = 1;
%WHILE, &whilevar = 1;                 /* "infinite" loop */
{
     %FONT, "System72.fnt";
     %MENU, @answername, 450,50, 1/*default choice*/,
          " POSSIBLE CHOICES: ",  /*in alphabetical order*/
          @08name,
          @04name,
          @03name,
          @16name,
          @07name,
          @06name,
          @12name,
          @14name,
          @02name,
          @01name,
          @13name,
          @09name,
          @11name,
          @15name,
          @10name,
```

```
    @05name,
    "----------------",
    "Quit the Quizzer";
%IF, @answername = "Quit the Quizzer";
{
    %EXECUTE, "Quizzer", 1;
}/* end IF */
%IF, @answername = @name;
{
     %FONT, "System72.fnt";
     %DISTEXTSTOP, &anykey, 180,350,
        9/*white on lt.blue*/,
        "  CORRECT ANSWER -- NICELY DONE!!  ",,
        "   (Press Q to Quit the Quizzer,   ",,
        "   M for More info on this image,  ",,
        "   or any other key to try again)  ";
     %CHOICE, &anykey;
     {
        %CASE= 77:  /* "M" in ASCII */
                    %EXECUTE, "Quizzer", 4;
        %CASE= 81:  /* "Q" in ASCII */
                    %EXECUTE, "Quizzer", 1;
        %DEFAULT:   %EXECUTE, "Quizzer", 2;
     }/* end CHOICE */
}/* end IF */
%ELSE;
{
     %FONT, "System72.fnt";
     %DISTEXT, &textwin, 143,325, 14/*black on yellow*/,
      "                                                ",,
      "  BZZZZZZT -- WRONG-O, FERTILIZER BREATH !!!  ",,
      "                                                ";
     &wronganswer = &wronganswer + 1;
     %IF, &wronganswer < 3;
     {
         %DISTEXT, &wronganswin, 178,400,
            14/*blk on yel*/,
          " That was attempt #",&wronganswer,
          "; try again... ";
         %WAIT,,3.0;
         %CLOSEWIN, &textwin;
         %CLOSEWIN, &wronganswin;
     }/* end IF */
     %ELSE;                  /* if &wronganswer = 3 */
     {
         %DISTEXT, &wronganswin, 153,400,
            14/*blk on yel*/,
          "  THAT WAS YOUR THIRD AND LAST ATTEMPT !!  ";
         %WAIT,,2.0;
         %CLOSEWIN, &textwin;
         %CLOSEWIN, &wronganswin;
```

```
                %DISTEXTSTOP, &anykey, 130,350,
                   13/*black on lt.magenta*/,
                "           THE CORRECT ANSWER WAS:          ",,
                "         ---> ", @name," <---          ",,
                "       (Press Q to Quit the Quizzer,      ",,
                "        M for More info on this image,    ",,
                "        or any other key to try again)     ";
                %CHOICE, &anykey;
                {
                   %CASE= 77:  /* "M" in ASCII */
                            %EXECUTE, "Quizzer", 4;
                   %CASE= 81:  /* "Q" in ASCII */
                               %EXECUTE, "Quizzer", 1;
                   %DEFAULT:   %EXECUTE, "Quizzer", 2;
                }/* end CHOICE */
             }/* end ELSE */
      }/* end ELSE */
}/* end WHILE */
END_OBJECT

/* 4: Processdoc for more info */
%CALL, "COMMON", 6;
%CALL, "COMMON", 2;
%FUNCTIONKEY, "ESC", "Quizzer", 2;
%PROCESSDOC, @image, "G", 1,1, @X,@Y,
     1/*color*/, @Xmin,@Ymin, @Xmax, @Ymax;
END_OBJECT

/* 5: Initialize variables */
%FONT, "System24.fnt";
%DISTEXT, &initwin, 245,240, 15/*black on white*/,
     " INITIALIZING... ";
@01name = "Su-24 FENCER      ";
     @01image = "SU24";
     @01X =      0; @01Y =      0;
     @01Xmin =  20; @01Ymin =  25;
     @01Xmax = 620; @01Ymax = 455;
@02name = "SOVREMENNYY DDG   ";
     @02image = "SOVREM_2";
     @02X =      0; @02Y =      0;
     @02Xmin =  20; @02Ymin =  25;
     @02Xmax = 620; @02Ymax = 455;
@03name = "KIEV CVHG Baku    ";
     @03image = "BAKU";
     @03X =    -15; @03Y =      0;
     @03Xmin = 150; @03Ymin =  90;
     @03Xmax = 515; @03Ymax = 390;
@04name = "IVAN ROGOV LPD    ";
     @04image = "IROGOV_1";
     @04X =      0; @04Y =      0;
     @04Xmin =  20; @04Ymin =  25;
```

```
     @04Xmax = 620; @04Ymax = 455;
@05name = "YANKEE Notch SSGN ";
     @05image = "YNOTCH_1";
     @05X =      0; @05Y =      0;
     @05Xmin =  20; @05Ymin =  25;
     @05Xmax = 620; @05Ymax = 455;
@06name = "Mi-28 HAVOC       ";
     @06image = "MI28";
     @06X =      0; @06Y =      0;
     @06Xmin =  20; @06Ymin =  25;
     @06Xmax = 620; @06Ymax = 455;
@07name = "Mi-24 HIND        ";
     @07image = "MI24";
     @07X =      0; @07Y =      0;
     @07Xmin =  20; @07Ymin =  25;
     @07Xmax = 620; @07Ymax = 455;
@08name = "BALZAAM AGI       ";
     @08image = "BALZAAM";
     @08X =      0; @08Y =      0;
     @08Xmin =  85; @08Ymin =  50;
     @08Xmax = 530; @08Ymax = 300;
@09name = "TARANTUL III PGG  ";
     @09image = "TARANTUL";
     @09X =    -10; @09Y =      0;
     @09Xmin =  20; @09Ymin =  25;
     @09Xmax = 620; @09Ymax = 455;
@10name = "Yak-38 FORGER     ";
     @10image = "FORGER_1";
     @10X =      0; @10Y =      0;
     @10Xmin =  50; @10Ymin =  30;
     @10Xmax = 590; @10Ymax = 330;
@11name = "TBILISI CV Tbilisi";
     @11image = "TBILISI1";
     @11X =      0; @11Y =      0;
     @11Xmin =  50; @11Ymin =  50;
     @11Xmax = 550; @11Ymax = 325;
@12name = "OSCAR SSGN        ";
     @12image = "OSCAR_2";
     @12X =      0; @12Y =      0;
     @12Xmin =  30; @12Ymin =  50;
     @12Xmax = 610; @12Ymax = 430;
@13name = "Su-27 FLANKER B   ";
     @13image = "FLANKER";
     @13X =      0; @13Y =      0;
     @13Xmin =  40; @13Ymin =  40;
     @13Xmax = 600; @13Ymax = 320;
@14name = "SLAVA CG          ";
     @14image = "SSN12";
     @14X =      0; @14Y =      0;
     @14Xmin =  20; @14Ymin =  25;
     @14Xmax = 620; @14Ymax = 455;
```

```
@15name = "Tu-160 BLACKJACK  ";
     @15image = "BLACKJ_1";
     @15X =      0; @15Y =      0;
     @15Xmin =  20; @15Ymin =  25;
     @15Xmax = 620; @15Ymax = 455;
@16name = "KIROV CGN         ";
     @16image = "KIROV_1";
     @16X =      0; @16Y =      0;
     @16Xmin =  20; @16Ymin =  25;
     @16Xmax = 620; @16Ymax = 455;
%CLOSEWIN, &initwin;
%RETURN;
END_OBJECT
```

U. COMMON.YOB

```
/* 1: MOUSE instructions */
%FONT, "System24.fnt";
%WRITE, 25,25, 15/*white ltrs*/,9/*lt.blue bkgnd*/,
     " Use Mouse or up/down arrow keys to highlight ",
     "desired selection. ";
%RETURN;
END_OBJECT

/* 2: PROCESSDOC instructions */
%CLEARSCREEN, 11/*lt.azure*/;
%FONT, "System24.fnt";
%WRITE, 2,20, 15/*white ltrs*/,9/*green bkgnd*/,
     " Use Mouse to select the item in the picture ",
     "you want to investigate. ";
%WRITE, 2,470, 15/*white ltrs*/,9/*green bkgnd*/,
     " Rt.mouse button or ESC = Previous screen    ",
     "F1 = Help    F2 = Browser ";
%RETURN;
END_OBJECT

/* 3: INVALID SELECTION error message */
%DISTEXT, &errormsgwin, 165,225, 4/*white on red*/,
     " Invalid selection; please try again. ";
%WAIT,, 3.0;
%CLOSEWIN, &errormsgwin;
%RETURN;
END_OBJECT

/* 4: EXIT confirmation */
%FONT, "System72.fnt";
%QUESTTXTWIN, &reply,
  " ARE YOU SURE YOU WANT TO EXIT? (y/n):  ", 145,225,
  1/*no.chars in reply*/, 1/*force uprcase*/, 4/*wt on red*/;
%IF, &reply = "Y";
{
     %END;
}
%ELSE;
{
     %RETURN;
}
END_OBJECT

/* 5: CLOSE WINDOWS */
%CLOSEWIN, @presswin;
%CLOSEWIN, @titlewin;
%RETURN;
END_OBJECT
```

```
/* 6: SET VARIABLES */
@disdoc = 1;
@clearscreen = 1;
%RETURN;
END_OBJECT
```

V. GENERIC.YOB

```
/* 1: GENERIC */
%IF, @clearscreen = 1;                    /* set by object above */
{
     %CLEARSCREEN, 11/*lt.azure bkgnd*/;
     @clearscreen = 0;
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> ", @callerobj,
     " --> GENERIC                              ";
%IF, @disdoc = 1;                         /* set by object above */
{
    @disdoc = 0;
}/* end IF */
%CALL, "COMMON", 1;                        /* mouse instructions */
%FONT, "System72.fnt";
%MENU, &sortvar, 183, 157, 1/*default choice*/,
     " LIST <whatever> SORTED BY:  ",
     "<designator? type?>",                  /* choice 1 */
     "<name?>              ",                /* choice 2 */
     "-------------------------",            /* separator */
     "[return to previous menu]",            /* choice 4 */
     "[exit GEOTREC]";                       /* choice 5 */
%CHOICE, &sortvar;
{
     %CASE= 1:   %EXECUTE, "GENERIC", 2;
     %CASE= 2:   %EXECUTE, "GENERIC", 2;
     %CASE= 4:   @clearscreen = 1;
                 %EXECUTE, @callerobj, 1;
     %CASE= 5:   %CALL, "COMMON", 4;
     %DEFAULT:   %CALL, "COMMON", 3;
}/* end CHOICE */
%EXECUTE, "GENERIC", 1;
END_OBJECT

/* 2: Menu, SORTED BY <whatever> */
%IF, @clearscreen = 1; /*screen cleared except 1st time thru*/
{
     %CLEARSCREEN, 11/*lt.azure bkgnd*/;
     %CALL, "COMMON", 1;                   /* mouse instructions */
}
%FONT, "System24.fnt";
%WRITE, 20,470, 15/*white ltrs.*/, 8/*dk.grey bkgnd*/,
     " MENU LEVEL:  Main --> ", @callerobj,
     " --> Generic --> SORT BY <whatever> ";
%FONT, "System72.fnt";
%MENU, &choicevar, 219,127, 5/*default choice*/,
     "      <generic>    ",
     " - SORTED BY <whatever> - ",
```

```
     "[return to previous menu]",    /* choice 2 */
     "[exit GEOTREC]",               /* choice 3 */
     "--------------------",         /* separator */
     " . ",                          /* choice 5 */
     " . ",                          /* choice 6 */
     " . ",                          /* choice 7 */
     "--------------------";         /* separator */
%FUNCTIONKEY, "ESC", "GENERIC", 2;
%CHOICE, &choicevar;
{
     %CASE= 2:                  /* "[Return to prev.menu]" */
               %EXECUTE, "GENERIC", 1;

     %CASE= 3:                  /* "[Exit GEOTREC]" */
               %CALL, "COMMON", 4;
               @clearscreen = 0;

     %DEFAULT:                  /* all other selections */
               %CALL, "GENERIC", 3;
               @clearscreen = 1;
}/* end CHOICE */
%EXECUTE, "GENERIC", 2;
END_OBJECT

/* 3: GENERIC CHOICE */
%CLEARSCREEN, 0/*black*/;
%FONT, "System72.fnt";
%DISTEXT, &tempwin, 78,200, 4/*white on red*/,
"                                                                    ",,
"SELECTION NOT YET AVAILABLE IN THIS PROTOTYPE APPLICATION",,
"                                                                    ";
%WAIT,,3.0;
%CLOSEWIN, &tempwin;
%RETURN;
END_OBJECT
```

W. GEOTREC.BAT

```
@echo off
cd c:\hyperdoc
metawndo
hyper logon
cls
cd c:\
@echo off
```

X. FULCRUM.BAT

```
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo                  GATEWAY to FULCRUM!!!
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo    -- If you do NOT want to start Fulcrum,
@echo          hold CTRL key down and press C; then at
@echo          DOS prompt [e.g., "C:\>"] type "geotrec"
@echo          and press ENTER;
@echo.
@echo    -- OTHERWISE,
@echo off
pause
call c:\proc\restart
:RESTART is a FULCRUM batch file, and as such is not
:provided in this Appendix
call c:\geotrec
@echo off
```

Y. PARADOX.BAT

```
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo                  GATEWAY to PARADOX!!!
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo    -- If you do NOT want to start Paradox 3.0,
@echo          hold CTRL key down and press C; then at
@echo          DOS prompt [e.g., "C:\>"] type "geotrec"
@echo          and press ENTER;
@echo.
@echo    -- OTHERWISE,
@echo off
pause
call c:\para
call c:\geotrec
@echo off
```

Z. PARA.BAT

```
@echo off
path=d:;c:;c:\msdos;c:\dbases\paradox3;
cd c:\dbases\paradox3
paradox3
cd c:\
path=d:;c:;c:\msdos;
@echo off
```

AA. INGRES.BAT

```
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo          GATEWAY to the THREAT DATABASE !!!
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo    -- If you do NOT want to start this application,
@echo         hold CTRL key down and press C; then at
@echo         DOS prompt [e.g., "C:\>"] type "geotrec"
@echo         and press ENTER;
@echo.
@echo    -- OTHERWISE,
@echo off
pause
call c:\threat
call c:\geotrec
@echo off
```

AB. THREAT.BAT

```
:THIS BATCH FILE SHOULD BE IN THE C: ROOT DIR, WITH INGRES
:PROGRAMS IN C:\INGRES AND SUBDIRS \BIN, \FILES, \LIB, \TMP,
:\DATA\{dbasename}, and \ABF\{dbasename}\{applicname}.
@echo off
path=d:;c:;\msdos;\ingres;\ingres\bin;
@echo off
cls
dbms -m %1 %2 %3 %4
run4gl threatdb c:\ingres\data\threatdb\sovietdb.img logon
cls
rmingres
path=d:;c:;\msdos;
@echo off
```

AC. HDTOOLS.BAT

```
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo             GATEWAY to HYPERDOC TOOLS!!!
@echo.
@echo NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
@echo.
@echo    -- If you do NOT want to start Hyperdoc Tools,
@echo         hold CTRL key down and press C; then at
@echo         DOS prompt [e.g., "C:\>"] type "geotrec"
@echo         and press ENTER;
@echo.
@echo    -- OTHERWISE,
@echo off
pause
call c:\hd
call c:\geotrec
@echo off
```

AD. HD.BAT

```
@echo off
cd c:\hyperdoc
metawndo
hdoutils
cls
cd c:\
@echo off
```

BIBLIOGRAPHY

Alexander, Michael, "Everyone's Talking Multimedia", *Computerworld*, vol.23, no.36, 4 September 1989.

Barrett, Edward, ed., *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information*, MIT Press, 1989.

Berk, Emily, and Devlin, Joseph, "Products: Hypertext Products", *PC Week*, vol.6, no.11, 20 March 1989.

Bonner, Paul, and Leong, Kathy Chin, "Quantum Leap: The Smart Document", *PC Computing*, vol.2, no.7, July 1989.

Bruno, Richard, "Interactive MultiMedia and the Man-Machine Interface", *CD-ROM EndUser*, vol.1, no.5, September 1989.

Coale, Kristi, "The Body Electric", *MacUser*, vol.5, no.3, March 1989.

Conklin, Jeff, "Hypertext: An Introduction and Survey", *IEEE Computer*, September 1987.

Crowston, Kevin, and Malone, Thomas W., "Intelligent Software Agents", *Byte*, vol.13, no.13, Dec 1988.

Defense Communications Agency (Defense Data Network Program Management Office), *DDN Data Defense Network*, Defense Communications Agency, Washington, D.C., 1984.

Fersko-Weiss, Henry, "Hypertext Connects Disparate Data: Extract a World of Data, Layer by Layer", *Lotus*, vol.5, no.3, March 1989.

Freedman, Alan, *The Computer Glossary*, The Computer Language Company, Inc., 1989.

Frentzen, Jeffrey. "Products: Text and Image-retrieval Software Packages", *PC Week*, vol.6, no.24, 19 June 1989.

Harrison, Shelley, "The ABCs of Multi-Media CD-I & DVI", *CD-ROM EndUser*, vol.1, no.5, September 1989.

Heid, Jim, "Getting Started with HyperCard.", *Macworld*, vol.6, no.5, May 1989.

Helgerson, Linda W., "Disseminating Digital Data: Why CD-ROM?", *CD-ROM EndUser*, vol.1, no.8, December 1989.

Hosinski, Joan M., "DOD Lab Studies Hypermedia as Boost for Existing Products", *Government Computer News*, vol.8, no.12, 12 June 1989.

Hutchison, Roger, "The ABCs of CD-ROM", *CD-ROM EndUser*, vol.1, no.5, September 1989.

Jarrett, Lynn, "Versatility Helps HyperCard Make Its Mark in Media", *Digital Review*, vol.6, no.12, 27 March 1989.

Jonassen, David H, "Designing Structured Hypertext and Structuring Access to Hypertext", *Educational Technology*, vol.28, no.11, Nov 1988.

Karon, Paul, "KnowledgePro Links Two Technologies; Software Hybrid Merges Expert System and Hypermedia", *PC Week*, vol.6, no.15, 17 April 1989.

Kellett, Daniel A., *Hypertext: Another Step Toward the Paperless Ship*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1989.

Keough, Lee, "On-Target Text Retrieval", *Computer Decisions*, vol.20, no.12, Dec 1988.

Leek, Matthew, "The MEDIA for CALS", *CD-ROM EndUser*, vol.1, no.8, December 1989.

Lehrer, Ariella, "HyperCard K-12: What's All the Commotion?", *Classroom Computer Learning*, vol.9, no.7, April 1989.

Leong, Kathy Chin, "Hypertext - the Ever-Growing Interconnected Whole", *PC-Computing*, vol.2, no.7, July 1989.

Leong, Kathy Chin, "Owl's Guide", *PC-Computing*, vol.2, no.7, July 1989.

Lewine, Donald, "Hypertext Made Hypereasy", *IEEE Software*, vol.5, no.6, Nov 1988.

Lind, David J., *Optical Laser Technology, Specifically CD-ROM, and Its Application to the Storage and Retrieval of Information*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1987.

Manchester, Phil, "Can Codd's Law Still Relate?", *Computer Weekly*, no.1138, 3 Nov 1988.

Marchionini, Gary, "Hypermedia and Learning: Freedom and Chaos", *Educational Technology*, vol.28, no.11, Nov 1988.

Halasz, Frank G., "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems", *Communications of the ACM*, vol.31, no.7, July 1988.

Miller, David C., "Digital MultiMedia Era Opens: Through a Glass, DMMly", *CD-ROM EndUser*, vol.1, no.6, October 1989.

Millman, Howard, "IBM Launches LinkWay", *PC-Computing*, vol.2, no.4, April 1989.

Morgan, Cynthia, "Toting Up Federal Micros: Analysis Finds 1.6 Million", *Government Computer News*, vol.9, no.1, 8 January 1989.

Naval Postgraduate School, Monterey, California, Department of Administrative Sciences, Working Paper No. 89-02, *Centralizing the Data, Distributing the Processing*, by Ran Giladi, and Moshe Zviran, January 1989.

Naval Postgraduate School, Monterey, California, Report NPS52-87-024, *Image Database Management In A Multimedia System*, by Klaus Meyer-Wegener, Vincent Y. Lum, and C. Thomas Wu, August 1988.

Naval Postgraduate School, Monterey, California, Report NPS52-87-050, *Integrating Advanced Techniques Into Multimedia DBMS*, by Vincent Y. Lum, C. Thomas Wu, and David K. Hsiao, November 1987.

Naval Postgraduate School, Monterey, California, Report NPS52-88-010, *Managing Multimedia Data - An Exploration*, by Klaus Meyer-Wegener, Vincent Y. Lum, and C. Thomas Wu, March 1988.

Nelson, Theodor H., "Replacing the Printed Word: A Complete Literary System", *IFIP Proceedings*, October 1980.

Oren, Tim, "The CD-ROM Connection: CD-ROMs, Hypertext and Tightly Structured Databases", *Byte*, vol.13, no.13, Dec 1988.

Orr, Joel N., "I Don't Know, but Xanadu (a Hypermedia System)", *Andrew Seybold's Outlook on Professional Computing*, vol.7, no.4, 31 October 1988.

Picarille, Lisa, "Hyperdoc Software Assists With Multimedia Presentations", *PC Week*, vol.6, no.9, 6 March 1989.

Picarille, Lisa, "Spinnaker Program Gives HyperCard Power to PCs", *PC Week*, vol.6, no.21, 29 May 1989.

Polmar, Norman, *Guide To The Soviet Navy*, 4th ed., Naval Institute Press, 1988.

Rawles, Richard, "Brown Delivers A-UX Version of Intermedia", *MacWEEK*, vol.3, no.18, 2 May 1989.

Rawles, Richard, "The Interface That Launched a Thousand Imitations", *MacWEEK*, vol.3, no.12, 21 March 1989.

Shaw, Richard Hale, "KnowledgePro Uses Hypertext, Expert System Rules to Build Applications", *PC Magazine*, vol.8 no.12, 27 June 1989.

Taylor, Nick G., "HyperShell Hypertext Control System (Product Description and Shareware Notice)", product documentation from Text Technology, Macclesfield, Cheshire, England, 1989.

Taylor, Robert Ryland, *Conversion of Hard-Copy Documents to Digital Format Utilizing Optical Scanners and Optical Storage Media*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1989.

Young, Andrew, "CD-ROM and UNIX", *CD-ROM EndUser*, vol.1, no.8, December 1989.

(no byline), "Erasable? CD-ROM? or Both?", *CD-ROM EndUser*, vol.1, no.5, September 1989.

(no byline), "GECI International Challenges Apple's Hypercard with Hyperdoc for MS-DOS Machines", *Computergram International*, no.1138, 17 March 1989.

(no byline), "IBM UK Launches LinkWay Hypertext Schools Language", *Computergram International*, no.1215, 7 July 1989.

(no byline), "Speedy Access With Hyperdoc Personal Toolkit", *Jane's Defence Weekly*, vol.11, no.11, 18 March 1989.

(no byline), "Verity Brings In Its New Approach to Text Retrieval for Unix, VAX-VMS, MS-DOS", *Computergram International*, no.1213, 5 July 1989.

INITIAL DISTRIBUTION LIST

		Number of Copies
1.	Library, Code 01422 Naval Postgraduate School Monterey, CA 93943-5002	2
2.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
3.	Mr. Corky Stradling (Code 421) Naval Ocean Systems Center San Diego, CA 92152-5000	2
4.	Barry A. Frew Administrative Sciences Department Code 54FW Naval Postgraduate School Monterey, CA 93943-5000	2
5.	LT Wayne F. Sweitzer USN Naval Intelligence Automation Center 4600 Silver Hill Road Washington, DC 20389-5000	2